

R:BASE 11

for Windows

Command Index



The Programmer's Guide to Building
R:BASE Databases and Custom Applications



R:BASE 11 for Windows

Command Index

by R:BASE Technologies, Inc.

All R:BASE commands are programming commands. Most can be used independently at the R> Prompt, some are used only in define mode, and those discussed in this chapter are generally used only in programs.

R:BASE 11 Command Index

Copyright © 1982-2025 R:BASE Technologies, Inc.

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

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

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

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

Trademarks

R:BASE®, Oterro®, 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

Table of Contents

Part I Command Index	14
1 Reading Command Syntax	15
2 Command Categories	15
3 A	17
ABORT	17
ALTER TABLE	17
APPEND	23
ATTACH	23
AUTOCHK	26
AUTONUM	29
4 B	32
BACKUP	32
BEEP	35
BREAK	36
BROWSE	36
BROWSE USING	40
5 C	42
CALL	42
CASCADE	42
CDECL	43
CHDIR	45
CHDRV	46
CHKDSK	46
CHOOSE	47
OPTION parameters.....	52
Title	54
List	55
Buttons	56
CLEAR	57
CLOSE	59
CLOSEWINDOW	59
CLS	60
CODELOCK	60
COMMENT	62
COMMENT ON	64
COMMIT	65
COMPUTE	66
CONNECT	68
CONTINUE	69
CONVERT	70
COPY	70
CREATE INDEX	72
CREATE SCHEMA	75
CREATE TABLE	76
CREATE VIEW	80

	CREATEOBJECT	83
	CROSSTAB	87
6 D	88
	DBCONN	88
	DEBUG	89
	DECLARE CURSOR	89
	DEFF	92
	DELETE	93
	DELETE DUPLICATES	94
	DELETEOBJECT	95
	DELROW	95
	DETACH	96
	DIALOG	96
	format	101
	OPTION parameters.....	106
	Message	107
	Input Field	107
	Buttons	109
	OK and CANCEL.....	109
	YES and NO	110
	Window	111
	Popup Menu	114
	LOOKUP	114
	DATE	116
	TIME	117
	DATETIME	119
	CALCULATOR.....	121
	DIR	122
	DISCONNECT	123
	DISPLAY	124
	DROP	125
	DUPROW	128
7 E	128
	EDIT	128
	EDIT USING	132
	ENTER	134
	ERASE	135
	EXIT	135
	EXITFORM	136
	EXPLAIN	136
8 F	138
	FCONVERT	138
	FETCH	139
	FILLIN	141
	FIRST	142
	FLUSH	142
	FORMS	143
9 G	143
	GATEWAY	143
	EXPORT	145
	DB (Paradox)	146
	DBF (dBASE)	148

	CSV (Comma Separated Values).....	149
	FIX (Fixed Width).....	151
	HTM (HTML)	153
	XLS (MS Excel Spreadsheet).....	155
	XLSW (MS Excel Workbook).....	158
	DOC (MS Word Document).....	162
	TAB (Tab Separated Values).....	164
	SLK (Symbolic Link).....	166
	DIF (Data Interchange Format).....	167
	WK1 (Lotus)	168
	WQ1 (QuattroPro).....	170
	SQL (SQL Script).....	172
	XML (Extended Markup Language).....	173
	CLBR (MS Windows Clipboard).....	175
	RTF (Rich Text Format).....	176
	SAV (SPSS Format).....	178
	PDF (Adobe Acrobat Portable Document).....	179
	LDIF (Data Interchange Format).....	181
	JSON (JavaScript Object Notation).....	182
	ODT (OpenOffice Text Document).....	183
	Export Dialog.....	185
	IMPORT	187
	DB (Paradox)	189
	DBF (dBASE)	190
	ASCII Text (*.txt, *.csv, *.tab).....	192
	XLS (MS Excel).....	194
	WKS (Lotus 1-2-3).....	196
	HTM (HTML)	198
	WKQ (QuattroPro).....	199
	XML (Extended Markup Language).....	201
	MDB (MS Access).....	203
	JSON (JavaScript Object Notation).....	205
	ODS (OpenOffice Spreadsheet).....	206
	Import Dialog.....	208
	GET	209
	GETPROPERTY	210
	GOTO	212
	GRANT	213
10 H	218
	HELP	218
11 I	219
	IF/ENDIF	219
	INPUT	221
	INSERT	221
	INTERSECT	223
12 J	225
	JOIN	225
13 L	226
	LABEL	226
	LAST	227
	LAUNCH	227
	LBLPRINT	230

	SCREEN	231
	PRINTER	233
	BMP	236
	DIF	237
	EMF	239
	ETXT	240
	GIF	241
	HTML	243
	JPG	248
	LOTUS	250
	PDF	252
	PNG	258
	QPRO	260
	RTF	261
	SVG	264
	SYLK	265
	TIFF	267
	TXT	268
	WMF	270
	XHTML	271
	XLS	277
	XLS2003	279
	XLSX	281
	EMAIL SET ON.....	282
	LIST	284
	LIST ACCESS	291
	LOAD	293
14 M	299
	MAXIMIZE	299
	MIGRATE	300
	MINIMIZE	300
	MKDIR	300
15 N	301
	NEWDB	301
	NEWPAGE	301
	NEWROW	301
	NEXTROW	302
	NEXTTAB	302
	NORMALIZE	302
	NULLSET	302
16 O	303
	ON CONNECT/PACK	303
	ON DISCONNECT/EXIT	304
	ON ERROR	305
	OPEN	306
	ORDER BY	307
	OUTPUT	308
17 P	311
	PACK	311
	PAUSE	313
	OPTION parameters.....	316
	Message	317

Button	318
Meter	318
Gauge Meter	319
PLUGIN	319
ExtForm	320
GetDBVersion.....	321
LoadDirectoryName.....	322
LoadFileName.....	323
LoadFileNamePlus.....	324
R:Calculator.....	325
R:Choose Builder Plugin.....	326
R:Dialog Builder Plugin.....	330
R:Dialog with Buttons Builder Plugin.....	333
R:MAPI Mail Plugin.....	336
R:Pause Builder Plugin.....	337
R:SysDialog Builder Plugin.....	341
R:Whois Plugin.....	343
SendKeys.....	344
SysDialog.....	346
TimeZone.....	348
TreeBuilder.....	349
PREVROW	351
PREVTAB	351
PRINT	352
SCREEN	353
PRINTER	355
BMP	358
DIF	359
EMF	361
ETXT	362
GIF	363
HTML	365
JPG	370
LOTUS	372
PDF	374
PNG	380
QPRO	382
RTF	383
SVG	386
SYLK	387
TIFF	389
TXT	391
WMF	392
XHTML	393
XLS	399
XLS2003.....	401
XLSX	403
EMAIL Parameter.....	404
PRNSETUP	406
PROCESSMESSAGE.....	411
PROJECT	413
PROPERTY	415
PUT	418

18 Q	419
QBE	419
QUERY	419
QUIT	421
19 R	422
RBAPP	422
RBBEDIT	422
RBDEFINE	423
RBEDIT	424
RBLABELS	424
RBSYNC	425
RECALC	426
RECOVER	428
REFF	428
RELOAD	429
RENAME	430
REPORTS	432
RESET	433
RESETROW	433
RESTORE	433
RETURN	435
REVOKE	436
RHELP	438
RHide	438
RMDIR	438
ROLLBACK	439
RPHONE	439
RSHOW	440
RULES	440
RUN	441
RUNRBA	443
20 S	443
SATTACH	443
SAVEROW	446
SCONNECT	446
SCREEN RESTORE	448
SDETACH	448
SDISCONNECT	449
SELECT	450
SELECT Functions.....	453
SELECT Analytical Functions	456
TOP	458
INTO	459
FROM	460
EXCEPT	462
LIMIT	462
INNER JOIN.....	463
OUTER JOIN.....	465
WHERE	467
Sub-SELECT.....	472
AS	473
GROUP BY	474

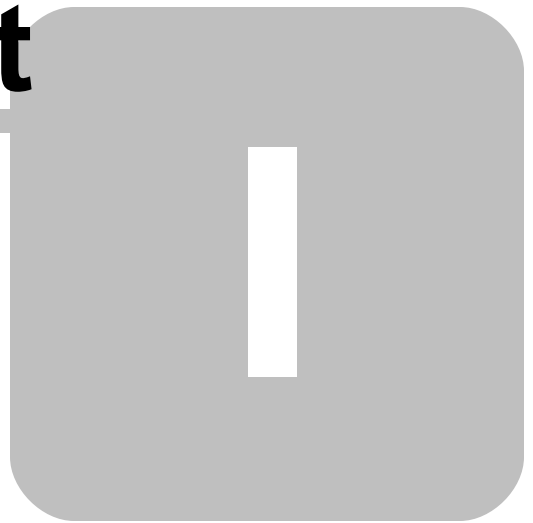
HAVING	476
ORDER BY	478
UNION	479
HTML	481
SET	482
AND	483
ANSI	483
ANSIOUTPUT.....	484
AUTOCOMMIT.....	484
AUTOCONVERT.....	484
AUTODROP.....	484
AUTORECOVER.....	485
AUTOROWVER.....	485
AUTOSKIP.....	485
AUTOSYNC.....	485
AUTOUPGRADE.....	486
BELL	486
BLANK	486
BOOLEAN.....	486
CAPTION.....	486
CASE	487
CHECKPROP.....	487
CLEAR	487
CLIPBOARD.....	487
CMDHIST	488
CMPAUSE.....	489
COLCHECK.....	489
COLOR	490
COMPATIB.....	490
CURRENCY.....	490
DATE	491
DEBUG	493
DELIMIT	493
ECHO	494
EDITOR	494
EOFCHAR.....	495
EQNULL	495
ERROR MESSAGE.....	496
ERROR MESSAGES.....	496
ERROR VARIABLE.....	497
ESCAPE	498
EXPLODE.....	498
FASTFK	499
FASTLOCK.....	499
FEEDBACK.....	500
FILES	500
FIXED	501
FONT	501
HEADINGS.....	501
IDQUOTES.....	501
INDEXONLY.....	502
INSERT	502
INTERVAL.....	502

KEYMAP	503
LAYOUT	503
LINEEND	503
LINES	503
LOCK	504
LOOKUP	505
MANOPT	505
MANY	505
MAXTRANS.....	505
MDI	506
MESSAGES.....	506
MIRROR	506
MOUSE	507
MULTI	507
NAME	507
NAMEWIDTH.....	508
NOCALC	509
NOTE_PAD.....	510
NULL	510
NUMROUND.....	510
ONELINE	510
PAGELOCK.....	511
PAGEMODE.....	512
PARSESEMI.....	512
PASSTHROUGH.....	513
PLUS	514
POSFIXED.....	514
PRINTER	514
PROCEDURE.....	515
PROGRESS.....	515
QUALCOLS.....	515
QUOTES	516
RBADMIN.....	516
RDEBUG	516
RECYCLE.....	517
REFRESH.....	517
REVERSE.....	518
ROWLOCKS.....	518
RULES	518
SCRATCH.....	518
SELMARGIN.....	519
SEMI	519
SEMI (Special Character).....	520
SERVER	520
SHORTNAME.....	520
SINGLE	521
SORT	521
SORTMENU.....	521
STATICDB.....	521
TIME	522
TIMEOUT	523
TJOURNAL.....	524
TOLERANCE.....	524

	TRACE	524
	TRANSACTION.....	525
	UINOTIF	525
	USER	526
	UTF8	527
	VERIFY	527
	WAIT	528
	WALKMENU.....	529
	WAREKI	529
	WHILEOPT.....	529
	WIDTH	529
	WINAUTH.....	530
	WINBEEP.....	530
	WRAP	531
	WRITECHK.....	532
	ZERO	532
	ZOOMEDIT.....	532
	SET STATICVAR	532
	SET VARIABLE	536
	SETFOCUS	540
	SHARECPY	540
	SHOW	541
	SHOW BUILD	544
	SHOW ERROR	545
	SHOW STATICVAR	546
	SHOW VARIABLE	547
	SHOW VERSION	548
	SKIP	548
	SSQL	549
	STDCALL	550
	SUBTRACT	552
	SWITCH/ENDSW	553
21 T	555
	TALLY	555
	TILE	556
	TRACE	557
	TURBO	558
	TYPE	558
22 U	559
	UNION	559
	UNLOAD	560
	UPDATE	568
	UPGRADE	571
23 W	572
	WHENEVER	572
	WHERE	573
	GROUP BY	579
	HAVING	581
	WHILE/ENDWHILE	583
	WRITE	586
24 Z	588
	ZIP	588

Part II Useful Resources	590
Part III Feedback	592
Index	594

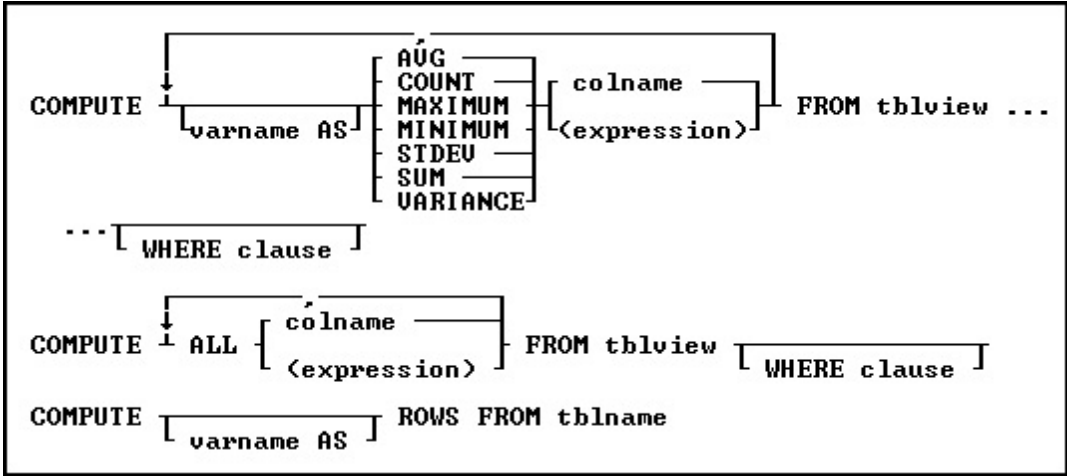
Part



1 Command Index

1.1 Reading Command Syntax

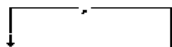
How To Read the Parts of the Command Syntax Diagram



Text Objects in the Command Syntax Diagram

- Keywords, which are uppercase, tell R:BASE what to do.
- Arguments, which are lowercase, represent specific information that you provide, such as a table name, column name, variable name, or expression.

Graphical Parts of the Command Syntax Diagram



An arrow in a command syntax indicates what portion of the command can be repeated. Each part of the command that is repeated must be separated with a comma, or the current delimiter character.

...

Ellipses indicate that the syntax continues to the next line.



If you have a choice of keyword or argument to use, the choices are enclosed in brackets.

This is the main line of the syntax. Any keywords or arguments on the main line are required.



This part of the syntax is below the main line and is therefore optional.

1.2 Command Categories

Configuration			
AUTONUM	MAXIMIZE	MINIMIZE	NORMALIZE
PRNSETUP	SET	SHOW	SHOW BUILD

[SHOW VERSION](#)**Control Structures**[BREAK](#)
[LABEL](#)
[WHILE/ENDWHILE](#)[CONTINUE](#)
[QUIT](#)[GOTO](#)
[RETURN](#)[IF/ENDIF](#)
[SWITCH/ENDSW](#)**Database Connection**[ABORT](#)
[ON CONNECT/PACK](#)[CONNECT](#)
[ON DISCONNECT/EXIT](#)[DBCONN](#)[DISCONNECT](#)**EEP Specific**[DELROW](#)
[LAST](#)
[PREVROW](#)
[SAVEROW](#)[DUPROW](#)
[NEWROW](#)
[PREVTAB](#)
[SCREENRESTORE](#)[EXITFORM](#)
[NEXTROW](#)
[RECALC](#)
[SKIP](#)[FIRST](#)
[NEXTTAB](#)
[RESETROW](#)**Error Handling**[DEBUG](#)[ON ERROR](#)[SHOW ERROR](#)[WHENEVER](#)**File Access**[ATTACH](#)
[INPUT](#)
[RUNRBA](#)
[SDISCONNECT](#)[CDECL](#)
[PLUGIN](#)
[SATTACH](#)
[STDCALL](#)[DETACH](#)
[REFE](#)
[SCONNECT](#)
[TYPE](#)[FLUSH](#)
[RUN](#)
[SDETACH](#)
[ZIP](#)**Object Manipulation**[CREATEOBJECT](#)[DELETEOBJECT](#)[GETPROPERTY](#)[PROPERTY](#)**Designer Modules**[DEFF](#)
[RBDEFINE](#)[FORMS](#)
[RBEDIT](#)[QBE](#)
[RBLABELS](#)[RBAPP](#)
[REPORTS](#)**Operating System**[CHDIR](#)
[COPY](#)
[MKDIR](#)[CHDRV](#)
[DELETE](#)
[RENAME](#)[CHKDSK](#)
[DIR](#)
[RMDIR](#)[CASCADE](#)
[ERASE](#)**Output Devices**[LBLPRINT](#)[OUTPUT](#)[PRINT](#)[WRITE](#)**Program Communication**[BROWSE USING](#)
[CLS](#)
[ENTER](#)
[PAUSE](#)
[RSHOW](#)[BEEP](#)
[DIALOG](#)
[EXIT](#)
[ProcessMessage](#)
[SETFOCUS](#)[CHOOSE](#)
[DISPLAY](#)
[FILLIN](#)
[RHELP](#)
[TILE](#)[CLOSEWINDOW](#)
[EDIT USING](#)
[NEWPAGE](#)
[RHIDE](#)**SQL / Query Language**[ALTER TABLE](#)
[CREATE INDEX](#)
[BROWSE](#)
[EDIT](#)
[INSERT](#)
[LOAD](#)
[QUERY](#)[APPEND](#)
[CREATE SCHEMA](#)
[DELETE](#)
[DROP](#)
[INTERSECT](#)
[OPEN](#)
[RENAME](#)[CLOSE](#)
[CREATE TABLE](#)
[DELETE DUPLICATES](#)
[FETCH](#)
[JOIN](#)
[ORDER BY](#)
[REVOKE](#)[COMMENT ON](#)
[CREATE VIEW](#)
[DECLARE CURSOR](#)
[GRANT](#)
[LIST ACCESS](#)
[PROJECT](#)
[RULES](#)

[SELECT](#)
[UPDATE](#)

[SSQL](#)
[WHERE](#)

[SUBTRACT](#)
[GROUP BY](#)

[UNION](#)
[HAVING](#)

Stored Procedures

[CALL](#)

[GET](#)

[PUT](#)

Transaction Processing

[COMMIT](#)

[RECOVER](#)

[ROLLBACK](#)

Utilities

[AUTOCHK](#)
[CONVERT](#)
[GATEWAY](#)
[MIGRATE](#)
[RBSYNC](#)
[RPHONE](#)
[TURBO](#)

[BACKUP](#)
[CROSSTAB](#)
[HELP](#)
[NEWDB](#)
[RELOAD](#)
[SHARECPY](#)
[UPGRADE](#)

[CODELOCK](#)
[EXPLAIN](#)
[LAUNCH](#)
[PACK](#)
[RESET](#)
[TALLY](#)
[UNLOAD](#)

[COMPUTE](#)
[FCONVERT](#)
[LIST](#)
[RBBEDIT](#)
[RESTORE](#)
[TRACE](#)

Variable Handling

[CLEAR](#)
[SHOW STATICVAR](#)

[NULLSET](#)
[SHOW VARIABLE](#)

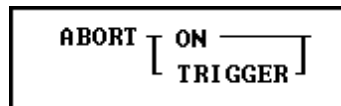
[SET STATICVAR](#)

[SET VARIABLE](#)

1.3 A

1.3.1 ABORT

Use the ABORT command in conjunction with the ON CONNECT/PACK command to stop the process, or with a stored procedure to stop a trigger.



Options

ON

The ON option is used in conjunction with the [ON CONNECT/PACK](#) command. It stops the connect or pack from being performed after the command file finishes.

TRIGGER

The TRIGGER option is used in a stored procedure to cancel the UPDATE, INSERT, or DELETE.

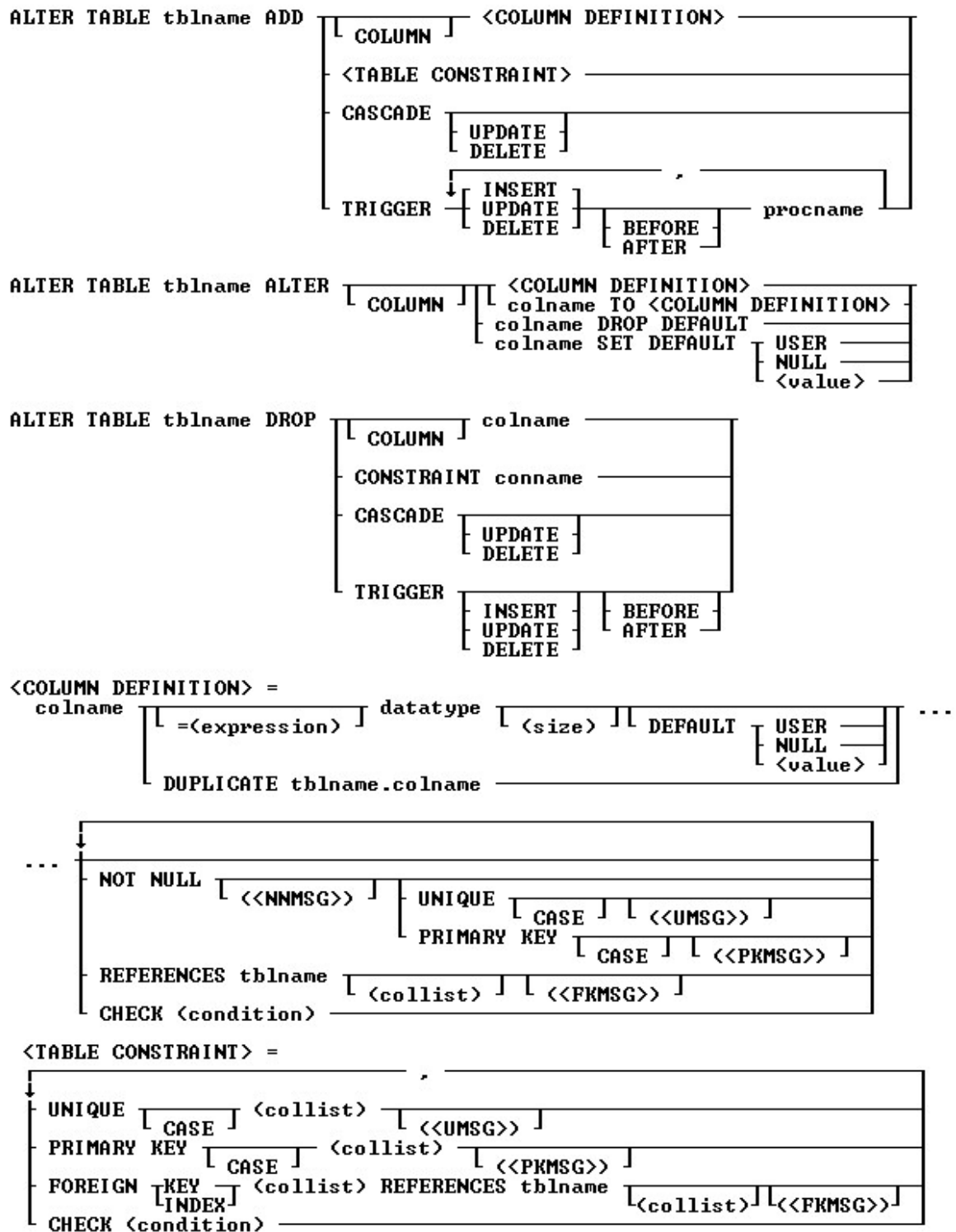
About the ABORT ON Command

The ABORT ON command is placed in the command file specified by the ON CONNECT/PACK command. It indicates that the CONNECT or PACK should not be performed after the command file finishes. For example, when using [AUTOCHK](#) with ON PACK to check the database for errors, use ABORT ON so the PACK is not performed if errors are found in the database.

Since a trigger runs a stored procedure before the row that triggered it is updated, inserted, or deleted, you can use the ABORT TRIGGER command to cancel the UPDATE, INSERT, OR DELETE.

1.3.2 ALTER TABLE

Use the ALTER TABLE command to modify an existing table.



Options

Indicates that this part of the command is repeatable.

ADD

Specifies the column and its definition, or a table constraint to add.

ADD CASCADE

Maintains primary/foreign key relationships automatically. For example, if you either UPDATE or DELETE a primary key value from a table, the corresponding foreign key values are updated or deleted automatically. A CASCADE can be applied to UPDATE, DELETE or BOTH to specific primary keys. By not specifying either UPDATE or DELETE, both CASCADE restrictions will be enforced upon the primary/foreign key tables. Separate UPDATE and DELETE data restrictions can allow a CASCADE to be enforced for records that are updated, but not enforced when records are deleted, in order to avoid an accidental or undesired record delete. CASCADE can only be added to tables with primary keys.

ADD TRIGGER

Adds the specified triggers to the table. Triggers run a stored procedure when an UPDATE, DELETE, or INSERT is executed. If you are using BEFORE and AFTER triggers, BOTH must be ADDED at the same time.

AFTER

Specifies the AFTER trigger event to activate or drop the INSERT, UPDATE or DELETE action.

ALTER

Modifies a column definition.

BEFORE

Specifies the BEFORE trigger event to activate or drop the INSERT, UPDATE or DELETE action. This is the default setting when creating a trigger, if the BEFORE/AFTER parameter is unused.

CASE

Specifies that the data values will be case sensitive.

CHECK (condition)

Sets a condition to be satisfied before an update or insertion of a row can occur, which creates an R:BASE rule.

(collist)

Specifies a list of one or more column names, separated by a comma (or the current delimiter), used in the unique key specification. This option is only used when referencing a unique key.

colname

Specifies a column name. The column name is limited to 128 characters.

COLUMN

Specifies the column to add, drop, or alter.

conname

Specifies a constraint name.

datatype

Specifies an R:BASE data type.

DEFAULT

Specifies a default value for the column if no value is provided by the user. The specified default value must match the column's data type (e.g. "PA" for a column that stores state address abbreviations). With TEXT columns, do not enclose the default value in the QUOTES character. The system variables #DATE, #TIME, and #NOW may be specified as default values. Static and global variables may also be specified as default values. When using system variables, static variables, or global variables, the value must be a dotted variable (the period must be included).

DROP

Removes a column or a constraint. A column, including both its structure and data, is removed from the table. Dropping a constraint removes a primary key, foreign key, unique key, or a not-null constraint.

DROP CASCADE

Disables the CASCADE feature so that primary/foreign key relationships are not maintained automatically.

DROP CONSTRAINT

Removes a constraint.

DROP DEFAULT

Removes a column's default value.

DROP TRIGGER

Drops triggers from a table. If the INSERT, UPDATE, or DELETE actions is not specified, all triggers are dropped from the table. If the BEFORE or AFTER events are not specified for an INSERT, UPDATE, or DELETE action, both BEFORE and AFTER triggers for the specified action are dropped from the table.

DUPLICATE tblname.colname

Copies the data type, length, default value, and NOT NULL flag if any. DUPLICATE does not support computed columns, and will not duplicate any comment/description for the column.

= (expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as #date, #time, and #pi.

(<FKMSG>)

Creates a constraint violation message to appear whenever a foreign-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a foreign-key constraint violation. You can define two messages: one for inserting and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

FOREIGN INDEX

With the FASTFK setting on, creates a foreign key that has an index using row pointers for data retrieval on selected columns.

FOREIGN KEY

Specifies a column or set of columns required to match values in a particular primary key or unique key constraint defined in a table.

FOREIGN KEY (collist)

If (collist) comprises one column, this option is equivalent to FOREIGN KEY. If two or more columns are included in (collist), the values in the listed columns must be unique as a group in each row. Each column must be separated by a comma (or the current delimiter).

(<NNMSG>)

Creates a constraint violation message to appear whenever a not-null data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a not-null constraint violation. A constraint must be dropped, then recreated in order to modify the violation message.

NOCHECK

Optional NOCHECK parameter does not update references to views, tables, and columns in forms, reports, labels, access rights, and rules. In this case, user assumes the responsibilities to update any references to views, tables, and columns in forms, reports, labels, access rights, and rules. This condition is ONLY available for the **ALTER COLUMN** command.

NOT NULL

Prevents a column from accepting null values, but permits it to accept duplicate values. If this option is specified without a setting for a default value, you cannot insert rows without specifying values for the given column.

(<PKMSG>)

Creates a constraint violation message to appear whenever a primary-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a primary-key constraint violation. You can define three messages: one for uniqueness, one for deleting, and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

PRIMARY KEY

Specifies the column(s) to designate as a primary key constraint.

PRIMARY KEY (collist)

If (collist) comprises one column, this option is equivalent to PRIMARY KEY. If two or more columns are included in (collist), the values in the listed columns must be unique as a group in each row. Only columns defined as not null can be included in (collist). Each column must be separated by a comma (or the current delimiter).

procname

The procedure name. If a procedure by this name already exists in the database, an error is generated.

REFERENCES

Identifies the primary key or unique key table to which the foreign key refers.

SET DEFAULT

Changes a column's default value.

(size)

Defines the length of a column of the TEXT data type (if not the default 8). Defines the precision and scale of a column of the DECIMAL or NUMERIC data type, if not the default of precision 9 and scale 0 (9,0). VARBIT, VARCHAR, and BIT either require or can have a size.

tblname

Specifies a table name. The table name is limited to 128 characters.

(<UMSG>)

Creates a constraint violation message to appear whenever a unique-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a unique-key constraint violation. You can define three messages: one for uniqueness, one for deleting, and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

UNIQUE

Requires the values in a column to be unique by creating a unique key constraint.

UNIQUE (collist)

If (collist) is one column, this option is equivalent to UNIQUE. If two or more columns are included in (collist), the values in the listed columns must be unique as a group in each row. Only columns defined as not null can be included in (collist). Each column must be separated by a comma (or the current delimiter).

USER**NULL****(value)**

Default USER: Specifies the default value to be the user identifier.

Default NULL: Specifies the default value to be null.

Default (value): Specifies the default to be the indicated value.

About the ALTER TABLE Command

ALTER TABLE creates a temporary internal table by copying a table's structure and data. You must have enough disk space to hold another copy of a table, and your database should not exceed the number of tables and columns R:BASE allows, which includes user-defined tables and system tables.

After the ALTER TABLE command has been executed, the temporary table goes away; however, the disk space the temporary table occupied is not available. To recover this space, pack or reload the database using the [PACK](#) or [RELOAD](#) commands.

The ALTER TABLE command is supported in EEPs.

Adding Columns

When you add a new column to a database, specify the name, data type, and length when the data type for the column is TEXT, or precision and scale when the data type for the column is DECIMAL or NUMERIC. When the column is computed, specify the name and expression—a data type is optional. When

the column already exists in the database, specify only the name-R:BASE uses the existing data type, and length, if applicable.

Database Access Rights with ALTER TABLE

When access rights for a table or view have been assigned using the [GRANT](#) command, ALTER TABLE requires the database owner's user-identifier or permission from the owner to alter specific tables.

Limitations of the ALTER TABLE Command

You cannot assign an index to a new column or transfer the index of an existing column with ALTER TABLE. If the added column should be indexed, use the [CREATE INDEX](#) command.

You also cannot add or transfer rules with ALTER TABLE. If you want a rule to apply to a column in the table, you must add it with the [RULES](#) command or use the Database Designer.

You cannot use ALTER TABLE to modify a view.

You cannot add a foreign key to a temporary table.

Examples

The following command adds mailadrs, a TEXT column 40 characters wide, at the end (or far right) of the customer table.

```
ALTER TABLE customer ADD mailadrs TEXT (40)
```

The following command adds the profit "computed" column at the end of product table. The value of profit is computed from the current row values for listprice multiplied by 1.05. The data type specified is REAL.

```
ALTER TABLE product ADD profit=(listprice * 1.05) REAL
```

The following command adds an "update only" cascade to the employee table.

```
ALTER TABLE employee ADD CASCADE UPDATE
```

The following command defines columns one through three as a case-sensitive primary key. Before you use this command, you must add a not-null constraint to each of the columns.

```
ALTER TABLE tablename ADD PRIMARY KEY CASE (column1, column2, column3) ('This is a message from the primary key')
```

The following command line adds a foreign index to the custid column and references the primary key in the customer table.

```
ALTER TABLE transmaster ADD FOREIGN INDEX (custid) REFERENCES customer
```

The following command drops the before insert trigger in the InvoiceHeader table.

```
ALTER TABLE InvoiceHeader DROP TRIGGER INSERT BEFORE
```

The following command line adds the test trigger for the SampleTriggers Table.

```
DROP PROCEDURE MySampleTrigger
PUT AFTER.PRC AS MySampleTrigger
ALTER TABLE SampleTriggers ADD TRIGGER INSERT AFTER MySampleTrigger
RETURN
```

1.3.3 APPEND

Use the APPEND command to copy rows from a table or view to the end of a table.

```
APPEND tblview TO tblname [ WHERE clause ]
```

Options

tblview

Names the table or view from which you want to copy rows-the source.

TO tblname

The name of the table to which you want to copy rows-the destination.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the APPEND Command

R:BASE only copies values from the source table or view that have matching column names in the destination table. Columns in the destination table that are not in the source table or view are filled with null values.

Rows are copied, not removed, from the source.

Example

The following command adds the rows containing new employee information from the newemp table to the end of emptable, a table containing information about previous employees. A WHERE clause is not specified, so all rows are copied to emptable.

```
APPEND newemp TO emptable
```

1.3.4 ATTACH

Use the ATTACH command to attach a dBASE file to an open R:BASE database.

```
ATTACH [ TEMPORARY ] filespec [ AS tablealias ] [ USING ndxlist ] [ ALIAS aliaslist ]
```

Options

ALIAS AliasList

To specify alias names for columns.

AS tablealias

Specifies an alias, or temporary name, for the dBASE table. A table alias is sometimes required when attaching files that do not follow the same table name restrictions as R:BASE.

filespec

A dBASE database name with a drive and path specification in the form D:\PATHNAME\FILENAME.

ndxlist

Specifies a list of index files to associate with the specified dBASE file. You do not have to include the extension for each index file. Separate index file names with a comma (or the current delimiter). Index files must be located with the specified dBASE file.

TEMPORARY

Allows you to create a temporary table with the ATTACH command. The temporary tables will disappear when the database is disconnected.

USING

Removes dBASE index files that were previously associated with the attached dBASE file when this option is used without a list of index files.

About the ATTACH Command

Before you can attach a dBASE file, an R:BASE database must be open. You can open an existing database or use the [CREATE SCHEMA](#) command to create a database.

Include the file specification when the file is located on a different drive or directory. You do not have to include the .DBF extension for the dBASE file.

When [STATICDB](#) is set on, which activates a read-only schema mode, ATTACH creates temporary dBASE tables. With STATICDB enabled, dBASE tables are created as temporary with or without the TEMPORARY keyword.

R:BASE directly reads and writes dBASE III and dBASE III PLUS data and index files. R:BASE can also read and write dBASE IV data files and index files that have the .NDX extension, just as dBASE III and dBASE III PLUS can share files with dBASE IV. R:BASE cannot read encrypted files nor read and write to dBASE IV index files, which have .MDX extensions.

Attaching to dBASE Files from a Network

From a network, R:BASE, dBASE III, and dBASE III PLUS users can access the same file at the same time. R:BASE can lock a dBASE file just as dBASE III and dBASE III PLUS can lock a dBASE file. When R:BASE is in multi-user mode, it does not support dBASE IV use. When a dBASE IV file is open by dBASE, R:BASE cannot access that file; when R:BASE attaches to a dBASE IV file, that file cannot be accessed by dBASE IV.

Listing dBASE Files

Use the LIST command to list the dBASE files in an R:BASE database. R:BASE displays DBF in the Rows column to indicate a dBASE file.

Reattaching dBASE Files to R:BASE

A dBASE file stays attached unless you use the [DETACH](#) command, which removes a dBASE file and its associated index file from the R:BASE database. The dBASE files stay attached because R:BASE remembers the attached dBASE files and index files when you open a database. At that time, R:BASE searches the current directory and path to find the attached dBASE file; therefore, the location of the dBASE files must be included in your path.

Associating and Modifying dBASE Index Files

You can associate a maximum of seven dBASE index files, which have .NDX extensions, with a dBASE data file by using the ndxlist option. R:BASE remembers each index file you associate with the dBASE data file. Also, use the ndxlist option to modify or preserve a set of indexes. If you issue another ATTACH command with a list of index files, R:BASE removes the current index files from the dBASE data file and associates the new list with the dBASE data file.

R:BASE updates the information stored in the dBASE data and index files each time you add or edit information in a dBASE file.

R:BASE Commands that Work with dBASE

The following R:BASE commands work with dBASE files. Limitations are noted following the table.

Commands that Work with dBASE			
ATTACH	DROP LABEL	LIST CURSORS	RENAME FORM

BROWSE	DROP REPORT	LIST DATABASES	RENAME OWNER
CHOOSE	DROP RULE	LIST FORMS	RENAME REPORT
COMMENT ON	DROP TABLE	LIST LABELS	RENAME VIEW
COMMIT (4)	DROP VIEW	LIST REPORTS	REPORTS
COMPUTE	EDIT	LIST RULES	REVOKE
CONNECT	EDIT DISTINCT	LIST TABLES	ROLLBACK (4)
CONTINUE	EDIT USING	LIST VIEWS	RULES
CREATE VIEW	ENTER USING	LOAD	SELECT
CROSSTAB	FETCH	OPEN CURSOR	SET
DECLARE CURSOR	FUNCTIONS	PACK (1)	SHOW
DELETE	GRANT	PRINT	TALLY
DELETE DUPLICATES	INSERT INTO	PROJECT (2)	UNLOAD (3)
DETACH	LBLPRINT	QUERY	UPDATE
DISCONNECT	LIST ACCESS	RBLABELS	WHENEVER
DROP CURSOR	LIST ALL	RELOAD	ZIP
DROP FORM	LIST COLUMNS	RENAME	COLUMN

1. dBASE files are not affected when you use a [PACK](#) command.
2. Using the PROJECT command, you can create a new table from an existing table from dBASE to R:BASE, but not from R:BASE to dBASE.
3. You can unload dBASE tables as ASCII only.
4. You cannot modify dBASE tables when transaction processing is on.

Note:

dBASE memo fields can be 64K in size. If the dBASE memo field is larger than 4K (the maximum size of an R:BASE note column), R:BASE reads as much as will fit. If you make changes and then write the record back to dBASE, the existing dBASE memo field is overwritten. The Carriage Return and Line Feed characters in dBASE are mapped to [Alt] + [0255].

R:BASE Commands that Do Not Work with dBASE

The following commands do not work with dBASE files in R:BASE.

Commands that Do Not Work with dBASE		
ALTER TABLE *	CREATE INDEX ON	JOIN
APPEND	CREATE SCHEMA AUTHOR	RENAME TABLE
AUTONUM	CREATE TABLE	RESTORE
BACKUP ALL	DROP COLUMN	SUBTRACT
BACKUP DATA	DROP INDEX	UNION
BACKUP STRUCTURE	INTERSECT	

* Column names for dBASE files can be changed with ALTER TABLE.

Examples

Example 01:

```
ATTACH SampGate USING CompID, ProdDesc
```

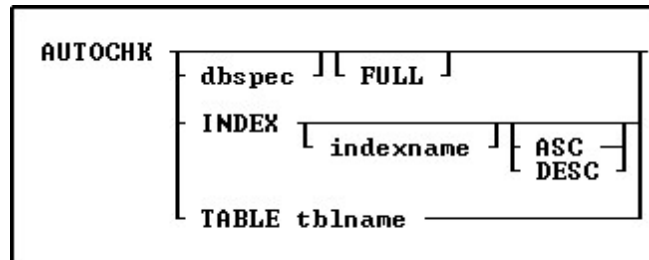
Example 02:

```
ATTACH TEMPORARY vendors.dbf +
AS VendorData +
```

ALIAS VendorName, VendorContact, VendorAddress, VendorPhone, VendorEmail

1.3.5 AUTOCHK

Use the AUTOCHK command to check the integrity of a database.



Options

ASC DESC

Specifies to check an index in ascending or descending order. Use DESC to check an index in descending order, by finding the last leaf and walking backwards in the B-tree index structure. The default index checking finds the first leaf and walks all the leaves in a forward direction.

dbspec

Specifies a database other than the open database to check; otherwise, the open database is checked.

FULL

Provides detailed information about the processing being performed, and when AUTOCHK encounters an error, it continues processing.

INDEX

Specifies to have all the indexes checked. The database must be connected in order to check indexes.

indexname

Specifies an index to check, which is displayed with the [LIST INDEX](#) command. The database must be connected in order to check an index.

TABLE tblname

Specifies a table name to check. The database must be connected in order to check a table.

About the AUTOCHK Command

Use the AUTOCHK command to ensure that the connected database is intact before using the [PACK](#) or [RELOAD](#) commands, or before making a backup of the database with either the [BACKUP](#) or [COPY](#) commands.

Please Note: If any user connected to the database has temporary tables or views created you may receive an abnormal amount of errors. This is expected and is a side effect of having temporary tables active during the check. For completely accurate results, have all users disconnect from the database that is being checked.

AUTOCHK checks the following:

- The structure-file block sizes and locations.
- The timestamps for all database files.
- The database-file lengths.
- The number of tables and columns.
- The starting and ending pointers for tables.
- The location of columns.
- The File 4 data pointers.
- The data types of columns.
- The size and number of rows in each table.

- The row pointers in the data file.

When you run AUTOCHK, it systematically checks the structure file of the open database, and the data files. AUTOCHK only checks the index file for the timestamp and length of the file. When opening a database, AUTOCHK ignores any user-identifier protection. AUTOCHK without the FULL option sets the R:BASE error variable to a non-zero value if errors are found.

The results of AUTOCHK with the FULL option are displayed on screen, or the current output device. First, AUTOCHK validates the timestamps in the database files, then systematically checks the structure of each table and view in the database, providing a list of columns, constraints, and indexes for each. Any structure errors are noted after each table listing.

AUTOCHK INDEX

The process reviews each index (file 3) and walks through every pointer to the data file (file 2). An error is reported if:

- The pointer from the index does not point to a valid location in the 2 file
- The pointer from the index points to a valid location in the 2 file, but the row that lives there has been deleted
- The row count from walking the entire index does not match the row count for the table as stored in file 1

Database Statistics

Next, AUTOCHK checks the data for each table, listing active rows and deleted rows. Any problems with data, such as broken pointers, are listed after the respective tables. Finally, AUTOCHK provides a summary of the database structure, including the number of tables, columns, and indexes, and the actual space that the data occupies in the data file (File 2). AUTOCHK shows the percent of space used for the items in each list to give an idea of how much space has been used, and to indicate the need to recover space in the database files. Any numbers less than 100 percent indicate the need to pack or reload the database using the [PACK](#) or [RELOAD](#) commands.

The following section contains information about using AUTOCHK in application files and capturing the error variables returned. This allows the application developer to prevent users from continuing to use a corrupted database.

```
SET ERROR VAR E1
WRITE 'Checking database for errors...'
AUTOCHK dbname
IF E1 > 40 THEN
    WRITE 'AUTOCHK has found errors in the database!'
    BEEP
ENDIF
If E1 > 0 and E1 < 50 THEN
    WRITE 'AUTOCHK will not run - User Abort or Out of Memory'
    BEEP
ENDIF
IF E1 = 0 THEN
    WRITE 'AUTOCHK successful - No errors found'
ENDIF
PAUSE 2
RETURN
```

If AUTOCHK with no option finds an error, it stops checking the database and displays one error message. If the error message (see list below) indicates that the database is damaged, you might want to start using a backup copy of the database. Alternatively, you might want to use R:SCOPE, a database repair tool available from R:BASE Technologies, Inc.

If AUTOCHK finds no errors, it displays the message "NO ERRORS FOUND." If you press any key while AUTOCHK is checking the database, the program stops and displays the message "USER ABORT." AUTOCHK automatically sets the error variable to the number corresponding to the message returned. For example, if the error "UNABLE TO OPEN DATABASE FILE 2" is returned, the error variable is set to 52.

Multi-User Databases

Use caution when running AUTOCHK in a multi-user environment. If the database being checked is currently open with **MULTI** set on, AUTOCHK places a database lock on the database. The database lock remains in effect until AUTOCHK stops checking the database. Database users are unable to make any changes to the data or structure of the database while this lock is in place.

If a user attempts to open a database being checked by AUTOCHK and the database does not have any other users, the user receives an error message indicating that the database is currently open in a mode that makes it unavailable. If other users have the database open with the **MULTI** set on and the database is being checked, the user attempting to open the database receives a message indicating the user is waiting in a lock queue. If AUTOCHK successfully completes checking the database and finds no errors, it reports that no errors were found and sets the error variable to 0.

Checking continues in multi-user mode (even if a database lock cannot be obtained) if a database is connected by another user; however, row errors in File 2 can occur because of database activity.

Error Messages

AUTOCHK displays one of the following messages when it is unable to start checking or complete checking the database or when it finds an error in the database files. AUTOCHK returns *0 No errors found* if the database is okay. Some of these messages indicate that the database is damaged. Either switch to a backup copy of the database, or attempt repair of the database using R:SCOPE, R:BASE Technologies's database repair tool. If AUTOCHK is unable to open File 1 of the database, check that the path you specified to the database is correct; or, if you are trying a multi-user database, check that no other user has the database connected with **MULTI** set off.

Checking continues in multi-user mode (even if a database lock cannot be obtained) if a database is connected by another user; however, row errors in File 2 might occur because of database activity.

Any of these messages, except the first (code 0), indicates that the database is damaged. Either switch to a backup copy of the database, or attempt repair of the database using R:SCOPE, R:BASE Technologies's database repair tool.

AUTOCHK Error Messages

Number	Code Message
0	No errors found
1	This database is not of the correct version
2	The database filenames must all match
20	Out of memory
40	User Abort
50	Unable to open database file number 1
51	Unable to lock this database
52	Unable to open database number 2
53	Unable to open database number 3
54	Unable to open database number 4
55	Error reading the database information block
56	Error reading the timestamp information
57	Timestamp in file number 2 does not match file 1; run RBSYNC
58	Timestamp in file number 3 does not match file 1; run RBSYNC
59	Timestamp in file number 4 does not match file 1; run RBSYNC
60	Invalid number of tables
61	Invalid number of columns
62	Invalid number of indexes
63	File 1 is too small
70	Error in database structure block
80	Error reading the table list
81	Error reading the column list
82	Error reading the index list
100	Incorrect version flag
101	Error reading Case Folding and Collating tables
110	Error in DBinfo block offset
111	Error in DBinfo block length
120	Error in length of database file 2

121	Error in length of database file 3
1609	Index rows counted: <value1>, expected count: <value2>
1610	The index points to an invalid row location
1611	The index points to a row that has been deleted
1612	The index cannot be examined

Examples

Example 01:

The following is an example of how to put AUTOCHK results in a file for viewing:

```
DISCONNECT
OUTPUT dbname.chk
AUTOCHK dbname FULL
OUTPUT SCREEN
```

You can view DBNAME.CHK in the R:BASE Editor to view the results.

Example 02:

The following checks the InvoiceDetails table structure and data:

```
AUTOCHK TABLE InvoiceDetails
```

Example 03:

The following checks the CustState index:

```
AUTOCHK INDEX CustState
```

1.3.6 AUTONUM

Use the AUTONUM command to define, modify, or remove an autonumber formula from a column.

```
AUTONUM colname IN tblname { DELETE
                              USING startnum [ increment ] [ format ] ...
... [ ORDER BY clause ] NUM
    NONUM
```

Options

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command.

DELETE

Removes a column's autonumber formula.

format

Defines the format in which values are displayed. This option is used only for columns with the TEXT data type.

You can use the following formatting characters:

<u>Formatting Character</u>	<u>Result</u>
9	Specify a numeric digit; leading zeros are suppressed.
0	Specify a numeric digit; leading zeros are displayed.
. (period)	Aligns digits along a decimal point.
[] (square brackets)	Encloses literal text.

For example, if the format is [MX]9999 and the numeric value is 123, the value entered will be MX123.

IN tblname

Specifies the table in which to autonumber the column.

increment

Specifies the value of the increment as each new row is added to the table. The default increment is 1.

NONUM

Leaves existing values unchanged and assigns autonumbered values to new rows as they are added to the table. NONUM is the default option.

NUM

Renumbers all the existing values in the column defined as an autonumbered column.

ORDER BY clause

Sorts rows of data. The [ORDER BY](#) clause is only used with the NUM option.

USING startnum

Defines or redefines the formula for an autonumber column. You must specify a starting value. Optionally, you can specify an increment, and for columns with the TEXT data type, a display format. For a column in a table that contains values, you can either renumber existing values or leave them as they are.

About the AUTONUM Command

An autonumbered column ensures that each row in that column has an incremental value. For example, use an autonumbered column to assign identification numbers, model numbers, or invoice numbers:

The following types of columns can be autonumbered:

- Columns that are not computed.
- Columns with DOUBLE, INTEGER, NUMERIC, REAL, or TEXT data types.

When you use the [LIST](#) command to list information about a column or table, autonumbered columns are described as AUTONUMBER in the *attributes* column.

Automatic Numbering

R:BASE automatically enters values in an autonumbered column when you add rows to a table using a form, the Data Editor, [INSERT](#) command, or [LOAD](#) command with the NUM option. When you import rows to a table that contains an autonumber column, you can either set autonumbering off and load imported values, or set autonumbering on and let R:BASE autonumber the values.

Capturing the Autonumbered Value

The next value for an autonumbered column can be captured for extended calculation or for display in a form. To capture the value, use the NEXT Function.

Changing Values

You can change the values in an autonumbered column by using a form, or the [UPDATE](#) or [EDIT](#) command. However, if you change a value in an autonumbered column, you could assign a duplicate number or disrupt the sequence of numbers. For more information about changing values in an autonumbered column, see the below guidelines in [Renumbering Columns Containing Data](#).

Renumbering Columns Containing Data

If you renumber a column that contains data, use the following guidelines to decide whether to change the column's existing values.

Autonumber?	Option to Use	Conditions
Yes	NUM	A column exists in only one table in the database. You can use the ORDER BY NUM clause to sort the rows in the order in which you want them renumbered. When you add new rows, values are numbered in the order in which the rows are added to the table.
No	NONUM (or do not specify)	A linking (or common) column exists in more than one table. You will destroy the common column values that link your tables if you renumber the values in a linking column. R:BASE adds autonumbered values to new rows as you add them to the table.

Redefining Formulas

You can redefine the formula for an autonumbered column. For example, use the AUTONUM command with the NUM option to change a column's display format from suppressing leading zeros to displaying them. For more information about redefining formulas of an autonumbered column, see "Renumbering Columns" earlier in this entry.

Removing Formulas

To remove an autonumber formula for a column, use the DELETE option. R:BASE removes only the formula, not the existing values in the autonumbered column. After you remove an autonumber formula, the user must enter values in the column as rows are added.

Autonumbering Tables Created with Relational Commands

When you create a table with one of the relational commands ([INTERSECT](#), [JOIN](#), [PROJECT](#), [SUBTRACT](#), or [UNION](#)), R:BASE transfers an autonumbered column as a regular column. You must define an autonumber formula for the column in the new table.

Database Access Rights with AUTONUM

When access rights for a table have been assigned using the [GRANT](#) command, AUTONUM requires either the database owner's user identifier, or the rights to alter a table.

Examples

The following command defines an autonumber formula for the *custid* column in the *customer* table. Existing values are renumbered starting at 100; assigned values increase by one for each row. Only use this command for a column that meets the renumbering guidelines in the section "Renumbering Columns Containing Data."

```
AUTONUM custid IN customer USING 100 1 NUM
```

The following command defines an autonumber formula for the *model* column in the *product* table. Existing values are not renumbered. Values in new rows are numbered starting with 100. Assigned values increase by one each time a row is added. The numbering format specifies that the letters MX always precede the numeric value. The 0000 provides space for a numeric value of up to four digits. When the value is less than four digits, R:BASE enters leading zeros.

```
AUTONUM model IN product USING 100 1 [MX]0000 NONUM
```

The following command assigns autonumbering to the *empid* column in the *employee* table. Existing values are renumbered starting at 100; assigned values increase by one for each row. The rows are

renumbered by the employees' last and first names. Only use this command for a column that meets the renumbering guidelines in the section "Renumbering Columns Containing Data."

```
AUTONUM empid IN employee USING 100 1 ORDER BY emplname, empfname NUM
```

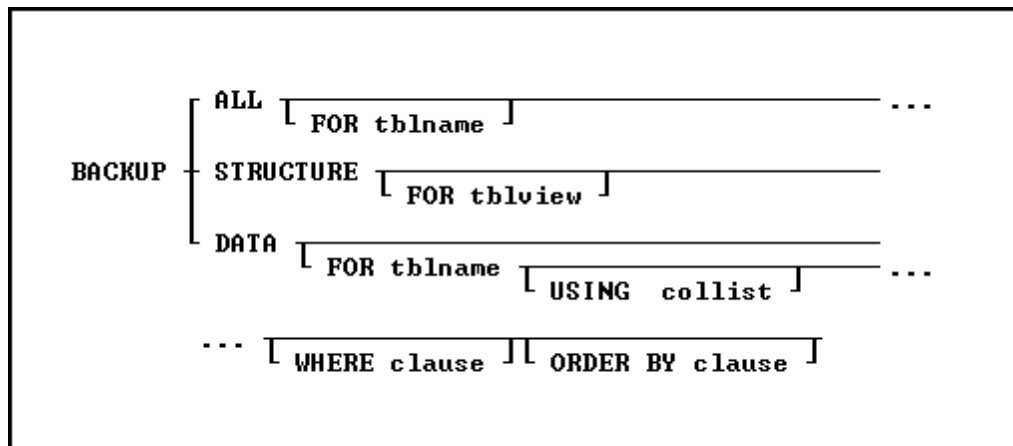
The command below deletes the autonumber formula from the *empid* column in the *employee* table.

```
AUTONUM empid IN employee DELETE
```

1.4 B

1.4.1 BACKUP

Use the BACKUP command to copy the data and/or structure of a database to floppy disks or another directory on your hard disk. The BACKUP command is very much like the [UNLOAD](#) command, only it supports the ability to span multiple diskettes when backing up databases to a floppy drive. With the many various backup software packages, or by simply performing file copies, the BACKUP command has become antiquated.



Options

ALL

Backs up the entire structure and data of the current database.

DATA

Backs up just the data of the current database.

FOR tblname

Specifies a single table to backup. If this clause is not used, all tables in the database are backed up.

FOR tblview

Specifies a table or view to backup. If this clause is not used, all tables and views in the database are backed up.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

STRUCTURE

Backs up just the structure of the current database.

USING collist

Specifies the columns to backup and in what order to back them up. USING ALL backs up all columns in the order they are defined in the table.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the BACKUP Command

The BACKUP command backs up data and/or structure to a backup file in the ASCII delimited format, which you can edit with a text editor. For example, you can restore the structure or data from one database to a new database, then edit the backup file with a text editor to change the database name to a new one.

BACKUP does not backup computed-column values; the values will be computed when the database is restored.

BACKUP does not change the data or the structure in the original database. Use the [RESTORE](#) command to restore your data.

If a BACKUP command is included in a transaction when transaction processing is on, the backup cannot be rolled back.

The BACKUP command creates a file with a .LOB extension for binary large objects, and a file for the data and/or structure.

SET Special Characters and Operating Conditions

To properly restore a database, BACKUP places the settings for the [SET](#) special characters and operating conditions used in the database at the beginning of the ASCII backup file. Within the backup file, object names, such as table names, will be enclosed within the IDQUOTES character. The current setting for DELIMIT is used to separate values. Text strings are enclosed in the current setting of QUOTES. And, the current value of WIDTH affects the width of data lines in the backup file. To view your database character and display settings, select "Settings" > "Configuration Settings" from the main Menu Bar.

The following are three tips that will ensure your database is restored from a backup properly:

- Set the null symbol to -0- (the R:BASE default) before backing up.
- Do not set a special character to the same setting as another special character.
- For best results backing up, keep all the default settings.

Backing Up to Disks

When you backup data to floppy disks, use blank formatted disks. The backup file can span multiple floppy disks. If BACKUP fills a disk, it prompts you to enter a new disk before continuing.

Note: So that you don't mix up the disks, label and number them as you remove them.

If you are backing up to a hard disk and the disk becomes full, press [Esc] to stop the backup. You can either free up hard disk space before backing up again, or backup to floppy disks.

It is worth noting that even when using R:BASE for Windows you will need to issue this command at the R> Prompt or via a command file.

Note: Backing Up to Floppy Disks is not recommended when using any 6.x, or higher, versions of R:BASE.

Using the BACKUP Command

When you backup your data, use the [OUTPUT](#) command to specify the filename of the backup. The BACKUP command does not erase files on a floppy disk, but overwrites a file if the filename you specify already exists on the disk. For examples of how to use the OUTPUT command, see the OUTPUT reference.

You can backup a database in one of the following ways:

- Backup both the structure and data.
- Backup only the structure.
- Backup only the data.

The following is information about each method.

Backing Up the Structure and Data

Use the BACKUP ALL command to backup the structure and data in the current database, or the structure and data for a specific table. You can backup a database structure and data to separate files if you prefer two smaller files to one large file. BACKUP writes all the commands necessary to define the database or table, starting with the [CREATE SCHEMA AUTHORIZATION dbname](#) command near the beginning of the file.

BACKUP uses the [LOAD](#) command to load the data.

If the database is protected by the database owner's user identifier, BACKUP ALL requires the owner's user identifier, which R:BASE places in the backup file to ensure that the restored database file continues to be protected. Keep the backup file from unauthorized users to ensure the database file remains protected.

Backing Up the Structure Only

Use the BACKUP STRUCTURE command to backup just the structure of the current database, or the structure of a specific table. BACKUP writes all the commands necessary to define the database or table, starting with the CREATE SCHEMA AUTHORIZATION dbname command near the beginning of the file. Any collating or case-folding table defined by the database owner is automatically preserved.

Keep the backup file from unauthorized users to ensure the database file remains protected. If the database is protected by an owner's user identifier, BACKUP STRUCTURE requires the owner's user identifier, which R:BASE places in the backup file to ensure that the restored database file continues to be protected.

Backing Up Data Only

Use the BACKUP DATA option to backup the data in the current database or the data in a specific table. BACKUP uses the LOAD command to load the data. Also, use the BACKUP DATA option to backup any custom information stored in R:BASE system tables such as forms, reports, labels and stored procedures.

Examples

The following command lines backup the structure and data for every table in the open database to the MYDB.BUP file on drive A:, redirect the output to the screen, and close the file.

```
OUTPUT a:mydb.bup
BACKUP ALL
OUTPUT SCREEN
```

The command lines below backup the data and structure of the customer table to the CUSTOMER.BUP file, limit the rows to those that are dated after 10/1/89, and order the rows by custid.

```
OUTPUT customer.bup
BACKUP ALL FOR customer WHERE datecol > 10/1/89 ORDER BY custid
OUTPUT SCREEN
```

The following command lines backup the structure of the customer table to the CUSTSTRC.BUP file, redirect the output to the screen, and close the file.

```
OUTPUT custstrc.bup
BACKUP STRUCTURE FOR customer
OUTPUT SCREEN
```

The following command lines backup the data from the producttable to the PRODUCT.BUP file, redirect the output to the screen, and close the file.

```
OUTPUT product.bup
BACKUP DATA FOR product
OUTPUT SCREEN
```

The following command lines show how to transfer a form to a new database. The first five command lines open the olddb database, set the null value to -0-, create the TEMPFORM.DAT file on drive D:, backup the data from the sys_forms3 table that contains the definition of the oldform form, and redirect the output to the screen. Command lines six through eight open the newdb database in order to transfer the oldform form definition (the database you open must not have a form named oldform), set the null value to -0-, and load the form definition from the backup file into the sys_forms3 table.

```
CONNECT olddb
SET NULL -0-
OUTPUT d:\tempform.dat
BACKUP DATA FOR SYS_FORMS3 WHERE SYS_FORM_NAME = 'oldform'
OUTPUT SCREEN
CONNECT newdb
SET NULL -0-
RUN d:\tempform.dat
```

1.4.2 BEEP

Use the BEEP command to play a beep sound or play an audio file. See also [WINBEEP](#).

BEEP ['FREQ <Frequency>,<Duration>' 'FILE Filename.wav']
--

Options

Duration

Time in milliseconds for the beep to sound.

FREQ

Allows for user specified beep properties. **Note:** This feature is only applicable to Windows 2000 and higher operating systems, and will be ignored by all other operating systems.

Frequency

A numeric value ranging from 37-32767 which represents the frequency of the beep in hertz.

FILE

Specifies the full path to a WAV audio file to play.

About the BEEP Command

BEEP is useful to signal when a process has finished, or to signal that an error has occurred in a command file.

Examples

Example 01:

-- Plays a beep sound

```
BEEP
```

Example 02:

-- Plays a beep sound at the 280 frequency for 1 second

```
BEEP 'FREQ 280, 1000'
```

Example 03:

-- Plays an audio file

```
BEEP 'FILE C:\Windows\Media\Windows Ding.wav'
```

1.4.3 BREAK

Use the BREAK command to force an early exit from a WHILE...ENDWHILE loop or a SWITCH...ENDSW structure.

BREAK

About the BREAK Command

The BREAK command is usually run in an [IF...ENDIF](#) structure contained within a WHILE...ENDWHILE loop or a CASE block within a SWITCH...ENDSW structure. The IF conditions indicate when to run the BREAK.

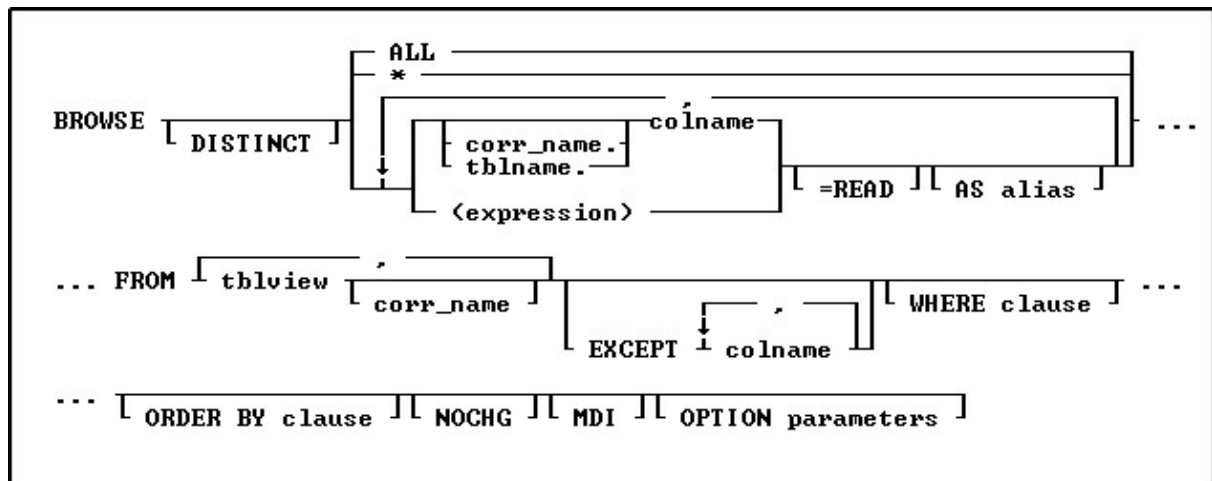
R:BASE exits the currently processing WHILE...ENDWHILE loop or SWITCH...ENDSW structure when a BREAK is encountered, and does not run any further commands in the WHILE loop or SWITCH structure. BREAK decreases the nesting level by one. BREAK passes control to the next line of the command file following the WHILE loop or SWITCH structure.

Examples

For an example of using BREAK with WHILE...ENDWHILE, see [WHILE...ENDWHILE](#). For an example of using BREAK with SWITCH...ENDSW, see [SWITCH...ENDSW](#).

1.4.4 BROWSE

The BROWSE command starts the Data Browser and displays the information you specify. With BROWSE, the data is not editable. To edit data with the Data Browser, use the [EDIT](#) command.



*
Specifies all columns.

,
Indicates that this part of the command is repeatable.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

ALL

Specifies all columns.

AS alias

Determines the alias of the column.

colname

Specifies a column name. The column name is limited to 128 characters.

corr_name.

Correlation name. A nickname or alias for a table or view name. Use *corr_name* to refer to the same table twice within the command, or to explicitly specify a column in more than one table. The correlation name must be at least two characters.

DISTINCT

Suppresses the display of duplicate rows.

EXCEPT colname

Specifies column names to be excluded from the Data Browser. The EXCEPT clause is helpful with tables which have many columns, to exclude columns from the result set. EXCEPT saves time to avoid typing a long list of column names. When using EXCEPT there can only be one table specified.

FROM tblview

Specifies the table or view.

MDI

Specifies a modeless Data Browser window, where other windows within R:BASE can be accessed without closing the current Data Browser window first. This option is used in R:BASE for Windows only.

NOCHG

Prevents changes to the issued BROWSE command and stops users from reaching additional data by adding columns to the command or changing the command's WHERE clause with the Update Query menu option. When the NOCHG option appears in the command, users cannot toggle from BROWSE to EDIT by pressing [F4], nor by using the **Edit** menu in R:BASE for Windows. NOCHG is particularly useful when the BROWSE command is issued within an application or command file.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

=READ

Specifies that the column is read-only, preventing changes to data.

tblname.

Specifies a column using the table name. Use *tblname.** to specify all columns in the table.

In a command, you can enter *#c*, where *#c* is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (*tblname.colname*). You can enter *tblname.** to specify all columns in the table.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

OPTION parameters - provides the ability customize the Data Browser Window. These enhanced parameters are supported at the R> Prompt, in command files, EEPs and stored procedures.

OPTION Parameters	Value	Description
CAPTION	value	specifies the window caption
CELL_HINTS	ON (Default) OFF	specifies if NOTE column cell hints are displayed
COLUMN_WIDTH[x] y	values	specifies the column width for a column, where X is the zero-based column index and Y is the width in pixels
CURRENT_FIELD_VALUE_VAR	value	specifies to capture the field value for the current focused cell
CURRENT_FIELD_VAR	value	specifies to capture the column name for the current focused cell
EXPORTRESULT	ON (Default) OFF	specifies if the "Export Result to->" menu item is available
GRIDVIEW		specifies the "View as Grid" mode is displayed

HEIGHT	value	specifies the window height, in pixels
IMAGE_HINT	ON (Default) OFF	specifies if image hints are displayed
LEFT	value	specifies the window location, in pixels, from the left of the screen over
LOCK	nnn	specifies to lock a number of columns
MODAL	ON (Default) OFF	specifies if the window is modal or non-modal. With the MODAL set to OFF, the Data Browser window displays as a separate button on the Windows task bar.
NO_FOCUS_FIELDS	value	specifies columns that cannot receive focus. The value is a comma separated list of columns.
ON_CELL_CLICK_EEP	value	specifies to execute the EEP command file when a cell is clicked
ON_CELL_DBL_CLICK_EEP	value	specifies to execute the EEP command file when a cell is double-clicked
PRINTDATA	ON (Default) OFF	specifies if the "Print Data" menu item is available
READONLY	ON OFF	specifies if the data is read only
ROW_NUMBERS	ON OFF (Default)	specifies if a record number is displayed (column to the far left)
ROWVIEW		specifies the "View as Row" mode is displayed
ROW_VIEW_DESCRIPTIONS	ON OFF (Default)	specifies to display the column comments from the table definition (when present) to the right of the data, when ROWVIEW mode is specified
SAVERESULT	ON (Default) OFF	specifies if the "Save Data As->" menu item is available
SAVERESULTTEMPONLY	ON OFF (Default)	specifies if the "Save Results As->" menu items are available for both permanent and temporary tables/views
TOP	value	specifies the window location, in pixels, from the top of the screen down
TREEGRIDVIEW		specifies the "View as Tree Grid" mode is displayed
UPDATE_QUERY	ON (Default) OFF	specifies if the query can be updated
WIDTH	value	specifies the window width, in pixels
WINDOW_STATE	MAXIMIZED NORMAL MINIMIZED	specifies the windows state
ZEBRA_STRIPE	ON OFF	specifies if zebra stripes are displayed

Examples

Example 01:

The following command displays all columns in the *transmastertable* and sorts the rows by the customer identification numbers contained in *custid* column.

```
BROWSE * FROM transmaster ORDER BY custid
```

Example 02:

The following command displays the *custid*, *company*, *custaddress*, *custcity*, *custstate*, and *custzip* columns in the *customer* table. Only the rows where the customer identification is equal to 100 are displayed.

```
BROWSE custid, company, custaddress, custcity, custstate, custzip FROM customer
WHERE custid = 100
```

Example 03:

The following command displays the *custid* column and restricts editing. Do not place a delimiter, such as a comma or a space, between options prefaced with an equal sign when you are building a command.

```
BROWSE custid=read, company, custaddress, custcity, custstate, custzip FROM customer  
WHERE custid = 100
```

Example 04:

The following command open all records for the Customer table with a maximized Data Browser window and "Customer Data" as the caption. Row numbers are displayed within the far left column.

```
BROWSE ALL FROM customer OPTION WINDOW_STATE MAXIMIZED|CAPTION 'Customer Data'|  
ROW_NUMBERS ON
```

Example 05:

The following will start the Data Browser for *custid* 125, displaying the data as a single row in read only mode.

```
BROWSE ALL FROM customer WHERE custid = 125 OPTION WINDOW_STATE NORMAL|ROWVIEW|  
READONLY ON
```

Example 06:

The following opens the Data Browser for the *Customer* table, with a limited height and width and located near the top left corner of the screen.

```
BROWSE ALL FROM Customer +  
OPTION WINDOW_STATE NORMAL +  
|TOP 25 +  
|LEFT 25 +  
|WIDTH 1000 +  
|HEIGHT 500
```

Example 07:

The following opens the Data Browser for active customers in the *Customer* table, with column alias names, a custom caption, and specified column widths for three that contain shorter data lengths.

```
BROWSE +  
CustID AS `Customer ID`, +  
Company AS `Company Entity`, +  
CustAddress AS `Home Office Address`, +  
CustCity AS `Home Office City`, +  
CustState AS `Home Office State`, +  
CustZip AS `Home Office Zip Code`, +  
CustPhone AS `Primary Phone Number`, +  
CustFax AS `Fax Number`, +  
LastUpdateDate AS `Last Date Modified`, +  
LastOrderDate AS `Last Order Date`, +  
CustURL AS `Company Home Page`, +  
CustEMail AS `Primary Email Address`, +  
CustStatus AS `Active Status`, +  
PaymentTerm AS `Payment Terms`, +  
CreditLimit AS `Imposed Credit Limit` +  
FROM Customer +  
WHERE CustStatus = 'A' +
```

```
ORDER BY Company ASC +
OPTION CAPTION 'Active Customer Information'+
|WINDOW_STATE NORMAL +
|COLUMN_WIDTH[4] 50 +
|COLUMN_WIDTH[5] 50 +
|COLUMN_WIDTH[12] 50
```

Example 08:

The following opens the Data Browser for contacts with email addresses in the *Contacts* table, and launched the CellDbClick.rmd command file when a cell is double clicked.

```
BROWSE CustID,ContFName,ContLName,ContEMail FROM Contact +
WHERE ContEMail IS NOT NULL +
OPTION ON_CELL_DBL_CLICK_EEP CellDbClick.rmd
```

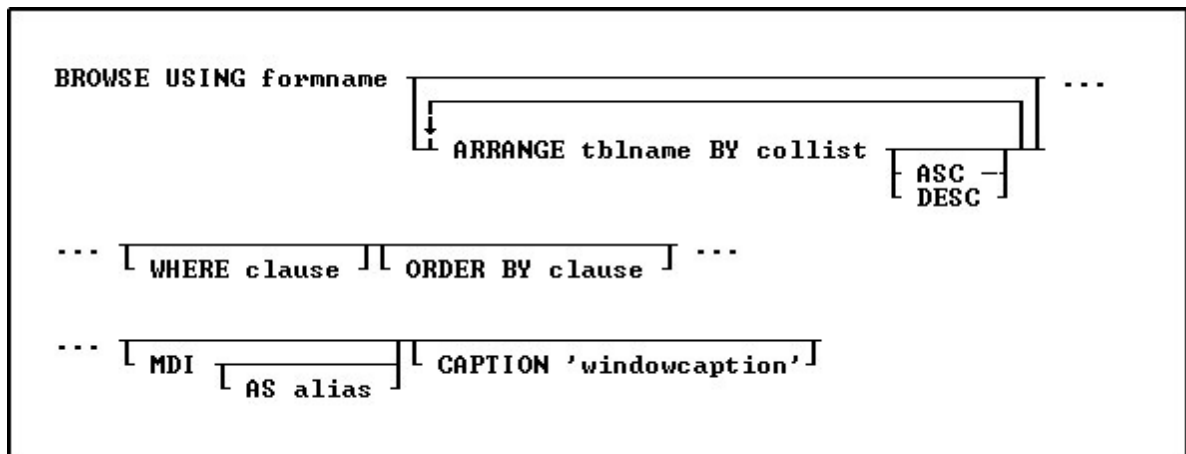
Example 09:

The following opens the Data Browser for California customers in the *Customer* table, and captures the column name and value wherever the focus is upon a cell.

```
SET VAR vCurFieldName TEXT = NULL
SET VAR vCurFieldValue TEXT = NULL
BROWSE CustID,Company,CustCity,CustState,CustZip FROM Customer +
WHERE CustState = 'CA' +
OPTION CURRENT_FIELD_VAR vCurFieldName +
|CURRENT_FIELD_VALUE_VAR vCurFieldValue
```

1.4.5 BROWSE USING

The BROWSE USING FormName command will make all column and variable objects on the form READ ONLY.



Options

formname

Specifies a form name. The form name is limited to 128 characters.

ARRANGE tblname BY collist

Sorts the rows displayed by a form's lower-level tables. Each lower-level table in the form that you want to sort requires its own ARRANGE clause. Now you can specify up to twenty-five tables to ARRANGE and up to five columns in each table on which R:BASE will sort.

**ASC
DESC**

Specifies whether to sort a column in ascending or descending order.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

MDI

This option is used in R:BASE for Windows only; specifies a modeless form-a modeless form allows you to access other windows without closing the form first.

AS alias

This option is used in R:BASE for Windows only. Allows the user to specify a name for this instance of the form. This option is used in conjunction with the MDI option so that the user can programmatically specify the form with the [SETFOCUS](#) command.

CAPTION 'windowcaption'

This option is used in R:BASE for Windows only. Specifies a caption for the window title bar.

About the BROWSE USING Command

The BROWSE USING FormName command will make all column and variable objects on the form READ ONLY.

Example:

```
BROWSE USING CustomerEdit WHERE CustID = 100
```

Notes:

- The Form Settings must include the option of "Editing Data".
- Only Column and Variable Objects are supported for READ ONLY mode when using BROWSE USING FormName command.
- Same Form could be used for entering and editing data accordingly.
- Any code defined within Entry/Exit Procedures (EEPs) will execute regardless of the status of the form in BROWSE mode, however, no table updates will be allowed.

The database RRBYW20 contains a five-table form called *SalesTransactions*. The command below shows how the form's four subordinate tables, *InvoiceHeader*, *InvoiceDetail*, *Employee* and *Contact*, can be sorted using an ARRANGE clause. The ARRANGE clause sorts the second table, *InvoiceHeader*, by the *TransID* column, in ascending order, sorts the third table, *InvoiceDetail*, by the *DetailNum* column, in ascending order, sorts the fourth table, *Employee*, by *EmpLName* and *EmpFName*, in ascending order and the fifth table, *Contact*, by the *ContLName* and *ContFName*, in ascending order.

```
CONNECT RRBYW20
```

```
BROWSE USING SalesTransactions +  
ARRANGE InvoiceHeader BY TransID ASC, +  
ARRANGE InvoiceDetail BY DetailNum ASC, +  
ARRANGE Employee BY EmpLName ASC, EmpFName ASC, +  
ARRANGE Contact BY ContLName ASC, ContFName ASC +  
ORDER BY CustID
```

1.5 C

1.5.1 CALL

The CALL command runs a stored procedure.

CALL procname([arglist])

Syntax

A) As function: SET VAR vVariable TYPE = (CALL procname(arglist))

B) As command: CALL procname(arglist)

About the CALL Command

The call command is used to invoke a stored procedure that was created by using the [SET PROCEDURE](#) command. It can be referenced by either a function notation or as a stand-alone command. In either case the argument list must be included. If you wish to include a blank argument list then use an empty pair of parenthesis.

Both methods of using CALL have their advantages. For example, using the function notation allows you to use a stored procedure in a computed view or to invoke a stored procedure via an SQL statement. On the other hand, used as a command, you will be able to reference the stored procedure by itself.

When using the Function notation the return value of the stored procedure is stored in the variable itself or displayed in the column (in the case of a computed column in a table or view). When using the Command notation the return value of the stored procedure will be placed into the system variable STP_RETURN. STP being an abbreviation of stored procedure.

The CALL command is supported in EEPs.

Examples

In the example below, a view is using a stored procedure to calculate values from another table.

```
CREATE VIEW MonthSum (CustomerID, CustomerSummary) +
AS SELECT T1.CustomerID, (CALL SumUpCust(T1.CustomerID)) FROM Customers T1
```

In the following example, an SQL select statement is used to invoke a maintenance routine from another application, written in Visual Basic, Java, or some other development environment. The use of WHERE LIMIT=1 causes the procedure to run once and only once. Without this clause the stored procedure would execute once for every matching row in the table. The *AnyTable* can be any table in the database in this case. The only requirement is that we must use a table in order to have a "healthy" [SELECT](#) clause.

```
SELECT (CALL DBCheck()) FROM AnyTable WHERE LIMIT=1
```

1.5.2 CASCADE

Use the CASCADE command to cascade all open windows. CASCADE was used in earlier Windows versions of R:BASE with MDI. The command is no longer supported.

CASCADE

1.5.3 CDECL

Use the CDECL command to call to a function or procedure as an external function declaration. Once the function or procedure name is declared, the DLCALL function may be used.

```

CDECL FUNCTION 'FunctionName' [ ALIAS 'AliasName' ] ...

... < PTR datatype [ <size> ] > : datatype [ <size> ]

CDECL VOID 'FunctionOrProcedureName' [ ALIAS 'AliasName' ] < PTR datatype [ <size> ] >

```

Options

ALIAS AliasName

Specifies an alias name for the function.

datatype

Specifies an R:BASE data type.

FUNCTION

Specifies that a function is declared.

FunctionName

Specifies the function name.

FunctionOrProcedureName

Specifies the function or procedure name.

PTR

Specifies a data type pointer.

(size)

Defines the length for the TEXT or VARCHAR data type. Defines the precision and scale of a column of the DECIMAL or NUMERIC data type, if not the default of precision 9 and scale 0 (9,0).

VOID

Specifies that the function does not return a value.

About the CDECL Command

CDECL is a declaration in the R:BASE environment to set a reference (access) to the function.

Calls from any DLL must be declared at least once in the R:BASE session from which they will be referenced. The CDECL declaration should be in a startup location for the R:BASE application. There is no limit to the number of declarations, but only make declarations that the application will use. Calling conventions specify how arguments are passed to a function, and how return values are passed back out of a function.

For best results, TEXT and VARCHAR data should be passed with the PTR (pointer) attribute. Any VARCHAR data type larger than 32K as a parameter, MUST be passed with the PTR attribute. It is important that the SIZE parameter on TEXT and VARCHAR be specified to avoid creating excess buffer space that has to be created from the declaration. The VARCHAR data type as a Return Value is restricted to 32K, but any SIZE up to 256MB can be passed as a pointer to the R:BASE variable. Modification of the data passed as pointer must be on the data pointed to.

CDECL and [STDCALL](#) are available as external function declaration commands. There is no performance difference between the two commands. When using CDECL, the calling program pushes the parameters on the stack and clears the stack on return. When using STDCALL, the calling program pushes the parameters on the stack but the called program clears the stack before doing the return. Choosing one over the other would depend on what style the DLL code being called is using.

Notes:

- The (size) parameter applies to TEXT and VARCHAR data types
- Parameters in the declaration are in the "reverse" order from the actual function or procedure in the DLL
- Parameters can be 0 to *n* parameters of any R:BASE data type
- Function names are case sensitive in the [STDCALL](#) or CDECL command declaration ONLY. The case must match the casing used in the DLL.
- The case is insensitive when used in the DLDCALL function.

Considerations:

If the function is declared as follows, then *nn* is less than or equal to 256 megabytes.

```
CDECL FUNCTION 'somefunction' ( PTR VARCHAR (nn)) : INTEGER
```

If the function is declared as follows, the buffer for VARCHAR will have default size equal to 256MB because the size is omitted. It is advised to **avoid** this declaration use, unless that is the actual size of the data to be passed.

```
CDECL FUNCTION 'somefunction' ( PTR VARCHAR ) : INTEGER
```

If the function is declared as follows, the buffer for VARCHAR will have default size equal to 32 kilobytes because the size is omitted.

```
CDECL FUNCTION 'somefunction' ( INTEGER ) : VARCHAR
```

If the function is declared as follows, *nn* bytes less than or equal to 32 kilobytes will be returned because the size is specified. When the size is specified, the data is passed as a parameter or as a return value will be truncated to the size value, if the data is greater than the size.

```
CDECL FUNCTION 'somefunction' ( INTEGER ) : VARCHAR(nn)
```

Examples:**Function Declarations:**

```
CDECL FUNCTION 'FunctionName' ( INTEGER ) : INTEGER
```

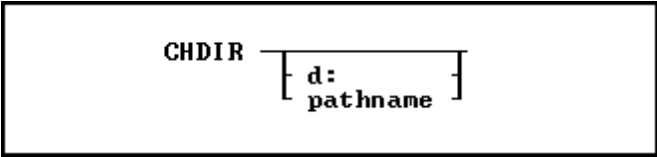
```
CDECL FUNCTION 'FunctionName' ALIAS 'FunctionAliasName' (PTR DOUBLE) : DOUBLE
```

Procedure Declarations:

```
CDECL VOID 'FunctionOrProcedureThatHasNoReturnValue' ( PTR TEXT (SIZE))
```


1.5.4 CHDIR

Use the CHDIR command to display or change the current directory.



```
CHDIR [ d:
      pathname ]
```

Options

d:

Specifies a valid drive letter.

pathname

Specifies one or more directory names that lead from the root directory to a specified directory. For paths with spaces, enclose the string in the apostrophe/single quote character.

About the CHDIR Command

R:BASE assumes you want to work in the current directory and drive. To display the current drive and directory, enter CHDIR without a path.

CHDIR is similar to the CHDIR command in the operating system; however, the R:BASE CHDIR command requires a space between the command and the directory. For more information about the CHDIR command, refer to your operating system manual.

CHDIR supports long (over 80 character) strings, and long strings with spaces.

Examples

Example 01:

The following command lines change the current directory to the SALEDEPT directory on drive C:.

```
CHDRV c:
CHDIR /saledept
```

Example 02:

The following command moves up one directory level and makes that directory the current directory.

```
CHDIR ..
```

Example 03:

At the R> Prompt or in a command file:

```
CD '\Documents and Settings\Administrator\Application Data\Microsoft\ +
  Internet Explorer\Quick Launch'
CHDIR
DIR
RETURN
```

Example 04:

At the R> Prompt or in a command file:

```
CD \FaxFiles\Archive\My_Very_Long_Corporation_Name\Service_Department\ +
  General_Pharmacy
CHDIR
```

DIR
RETURN

1.5.5 CHDRV

Use the CHDRV command to change the current drive.

`[CHDRV] d:`

Options

d:
Specifies a valid drive letter.

About the CHDRV Command

R:BASE assumes you want to use the current drive.

To display the current drive, use CHDRV without designating a drive.

Examples

The following command displays the current drive.

```
CHDRV
```

The following command makes drive A: the current drive.

```
CHDRV a:
```

The following command makes drive C: the current drive.

```
c:
```

1.5.6 CHKDSK

Use the CHKDSK command to display information about an active local drive.

`CHKDSK [d:
FH]`

Options

d:
Specifies a drive other than the current drive.

FH
Displays the number of available file handles.

About the CHKDSK Command

CHKDSK lists the following:

- The total number of bytes of disk space.
- The number of bytes of disk space available for use.

To display information about another drive-not the current drive-specify the drive.

CHKDSK is similar to the CHKDSK command in the operating system; however, the operating system command can correct errors in the disk directory or file allocation table.

Use CHKDSK to check available file handles after opening a database, executing a command, running a file, or printing reports. In R:BASE, a maximum of 20 file handles can be open at one time. R:BASE uses four file handles for an open database, two file handles to sort information, two file handles for a temporary scratch file, one file handle for files, and one file handle for each of the following commands: [RUN](#), [DISPLAY](#), [CHOOSE](#), [OUTPUT](#) *file name*, and [TYPE](#).

Examples

The following command displays the total bytes available on the current disk.

```
CHKDSK
```

The display looks similar to the lines below.

```
10592256 bytes total disk space
774144 bytes available on disk
```

The following command checks the total disk space available for the disk in drive A.

```
CHKDSK a:
```

The following command displays the number of file handles currently available.

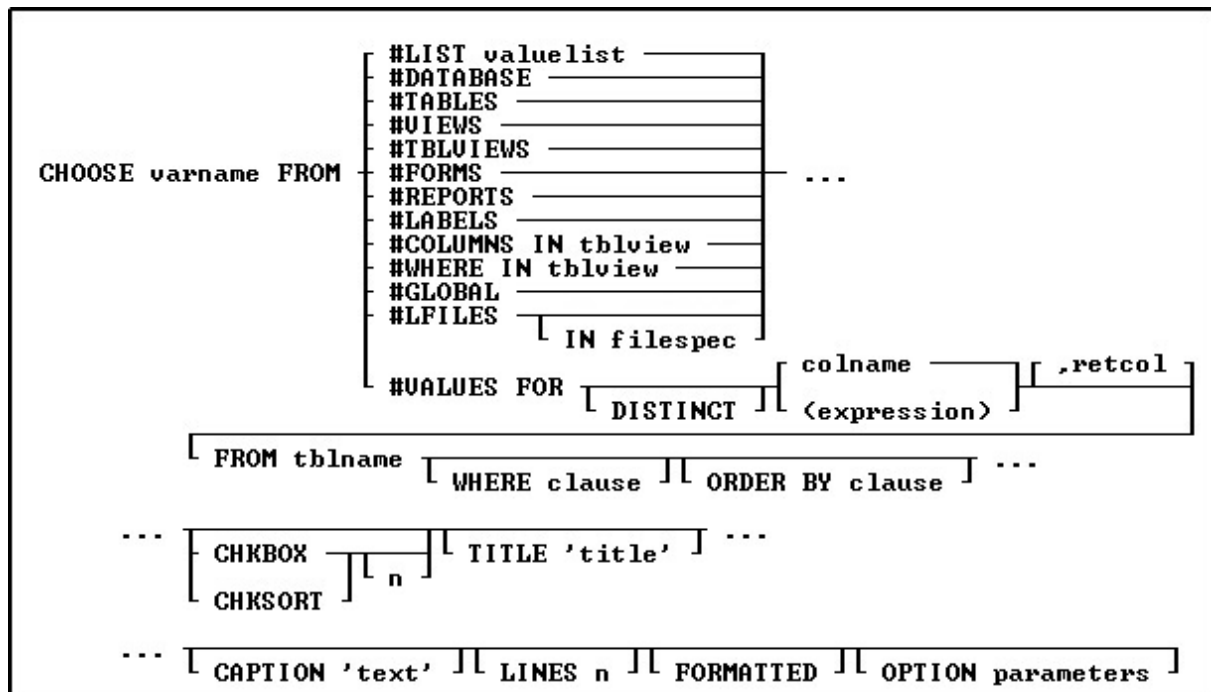
```
CHKDSK FH
```

The display looks similar to the lines below.

```
File handles: 9 10 11 12 13 14 15 16 17 18 19
```

1.5.7 CHOOSE

Use the CHOOSE command to display the following types of menus: bar with a pull-down, check-box, pop-up, sort check-box, vertical, or WHERE builder. For assistance with building your CHOOSE commands, refer to the [R:Choose Builder Plugin](#).



Options

#COLUMNS IN tblview

Displays a pop-up menu containing all columns in the specified table or view and stores the chosen column name in the variable varname.

#DATABASE

Displays a pop-up menu containing all databases in the current directory, and stores the chosen database name in the variable varname.

#FORMS

Displays a pop-up menu containing all forms in the open database and stores the chosen form name in the variable varname.

#GLOBAL

Displays a pop-up menu containing currently defined global variables and stores the chosen variable name in the variable varname.

#LABELS

Displays a pop-up menu containing all labels in the open database and stores the chosen label name in the variable varname.

#LFILES IN filespec

Displays all files in the current directory in a pop-up list, unless the IN filespec option is specified. In this latter case, the user may specify a filespec such as *.RTF, and R:BASE will display a pop-up list of all files in the specified directory with the .RTF extension. The chosen file name is stored in the variable varname.

#LIST valuelist

Allows you to specify a list of values in a comma delimited format. You can also use a variable that contains comma delimited values. The group of values MUST be encapsulated in quotes UNLESS a variable is used.

#REPORTS

Displays a pop-up menu containing all reports in the open database and stores the chosen report name in the variable varname.

#TABLES

Displays a pop-up menu containing all tables in the open database and stores the chosen table name in the variable varname.

#TBLVIEWS

Displays a pop-up menu containing all tables and views in the open database and stores the chosen table or view name in the variable varname.

#VIEWS

Displays a pop-up menu containing all views in the open database and stores the chosen view name in the variable varname.

#WHERE IN tblview

Opens the WHERE builder for you to enter conditions for the columns in the specified table or view; stores the entire WHERE clause in the variable varname. Your [WHERE Clause](#) conditions can also pre-loaded into the WHERE Builder using the **OPTION WHERE_CLAUSE** parameter.

#VALUES FOR

Displays a pop-up menu containing values from the specified column or expression and table. Use the (expression) option to display more than one column from the table in the menu, such as (colname1 & colname2).

DISTINCT

Suppresses the display of duplicate rows.

colname

Specifies a column name. In a command, you can enter #c, where #c is the column number shown when the columns are listed with the LIST TABLES command. In an SQL command, a column name can be preceded by a table or correlation name and a period (tblname.colname).

<expression>

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as #DATE, #TIME, and #PI.

,retcol

Specifies the column whose value is returned into the variable varname. The column does not have to be one of the columns displayed in the menu.

FROM tblname

Specifies the table name.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

CHKBOX

Displays a menu that limits the number of selections the end user can make.

CHKSORT

Displays a menu that limits the number of selections the end user can make, with support to sort the selected results.

n

n is an optional positive integer specifying the maximum number of options on that menu that can be checked. If n is zero or is greater than the number of menu options, all options can be checked. If n is unspecified, the default value is zero. The maximum value of n is 9999.

TITLE 'title'

Displays a title in the dialog box. The title is limited to 150 characters.

CAPTION 'text'

Displays text in the dialog box title bar

LINES n

Determines the number of lines, *n*, to display in the list box. The default is 10.

FORMATTED

Displays the CHOOSE box using a mono space font

Additional [OPTION parameters](#)

Additional parameters are available to increase the visual display of the CHOOSE window. To use the graphic CHOOSE Builder, choose "Utilities" > "Plugins" > "CHOOSE Builder" from the main Menu Bar. All OPTION parameters and values must be separated by the "|" (pipe) character.

Notes:

- CHOOSE will always be displayed at the center of your screen, unless specified using the TOP *nn* LEFT *nn* parameters.
- The resulting variable will always be TEXT
- The resulting variable will be left justified without leading spaces.
- The [F2] key is recognized when the CHOOSE is displayed and behaves similar to the [Enter] key. Instances to use [F2] would be if no items in the CHOOSE would apply, and the user does not wish to close the CHOOSE with [Esc]. The [F2] status can then be captured with the LASTKEY function for additional program control.

Example 1:

```
CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 01' LINES 20 FORMATTED
IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN
```

Example 2:

```
CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
CHKBOX 1 TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 02' LINES 20 FORMATTED
IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN
```

Example 3:

```

CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 03' LINES 20 FORMATTED +
OPTION List_Font_Color WHITE|List_Back_Color TEAL +
|Title_Font_Color TEAL|Title_Back_Color WHITE +
|Window_Back_Color WHITE|Title_Font_Size 24 +
|Title_Font_Name TIMES NEW ROMAN|Window_Caption
IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN

```

Example 4:

```

CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 04' LINES 20 FORMATTED +
OPTION List_Font_Color WHITE|List_Back_Color TEAL +
|Title_Font_Color TEAL|Title_Back_Color WHITE +
|Window_Back_Color WHITE|Title_Font_Size 24 +
|Title_Font_Name TIMES NEW ROMAN|Window_Caption
IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN

```

Example 05:

```

CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
CHKBX 1 TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 05' LINES 20 FORMATTED +
OPTION List_Font_Color WHITE|List_Back_Color TEAL +
|Title_Font_Color TEAL|Title_Back_Color WHITE +
|Window_Back_Color WHITE|Title_Font_Size 24 +
|Title_Font_Name TIMES NEW ROMAN|Window_Caption +
|Buttons_Back_Color WHITE

```

```

IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN

```

Example 6:

```

CLS
CHOOSE vCustIDTxt FROM #VALUES FOR +
(LJS(Company,40)&CustCity+', '&CustState),CustID +
FROM Customer ORDER BY Company +
CHKBOX 1 TITLE 'Company Name and Location' +
CAPTION 'Colorful CHOOSE Example 06' LINES 20 FORMATTED +
OPTION List_Font_Color WHITE|List_Back_Color TEAL +
|Title_Font_Color TEAL|Title_Back_Color WHITE +
|Window_Back_Color WHITE|Title_Font_Size 24 +
|Title_Font_Name TIMES NEW ROMAN|Window_Caption +
|Buttons_Show_Glyph ON|Buttons_Back_Color WHITE
IF vCustIDTxt IS NULL OR vCustIDTxt = '[Esc]' THEN
    GOTO Done
ELSE
    SET VAR vCustID = .vCustIDTxt
ENDIF
__ Do what you have to do here with the variable CustID
LABEL Done
CLEAR VAR vCustIDTxt, vCustID
RETURN

```

Example 7:

```

CLS
CHOOSE vWhere FOR #WHERE IN Customer +
OPTION THEMENAME R:BASE Rocks! +
|WHERE_CLAUSE WHERE CustState = 'CA' ORDER BY COMPANY
RETURN

```

1.5.7.1 OPTION parameters

These parameters can be used to change the "look and feel" of a CHOOSE window. In addition to the parameters below, you can also alter the CHOOSE [Title](#), [List](#), and [Buttons](#).

Parameter	Value	Description
WINDOW_CAPTION	OFF SMALL	Customizes the window caption. The OFF value makes window caption invisible. The SMALL value makes window caption small (tool window).
WINDOW_BACK_COLOR	value	Changes the color of the window background area. User can specify the integer value or type one of the predefined values.
TOP	value	Moves the CHOOSE box location, in pixels, from the top of the screen down.
LEFT	value	Moves the CHOOSE box location, in pixels, from the left of the screen right.

WHERE_CLAUSE	value	Allows you to pre-load a WHERE Clause into the WHERE Builder window when using "#WHERE IN tblview".
SHOW_ALL_BUTTON	ON OFF	Displays a "Select All" button when using the CHKBOX option.
SINGLE_CLICK	ON OFF	Limits the user to select a single option from the displayed list, and then immediately close. This option works only when user can select single item from a list. For multiple item selection this feature will be ignored.
INITPOS	value(s)	Specifies the preselected initial position(s) for the items in the list. The first value in the list is assigned zero, and increments for the number of values in the list.
INITVAL	value(s)	Specifies the preselected initial value(s) for the items in the list
THEMENAME	value	Specifies one of 86 pre-defined Themes, or a custom Theme loaded into R:BASE. See Themes. NOTE: All previously defined CHOOSE commands which are redefined to use Themes should be thoroughly checked prior to putting into production as objects and text may look substantially different.

```
-- Example Using INITPOS Option
IF (CVAL('DATABASE')) <> 'RRBYW20' OR (CVAL('DATABASE')) IS NULL THEN
    CONNECT RRBYW20 IDENTIFIED BY NONE
ENDIF
SET VAR vTitle TEXT = 'Choose Table(s)'
SET VAR vCaption TEXT = 'Using INITPOS Option in CHOOSE Command'
SET VAR vChoose TEXT = NULL
CLS
CHOOSE vChoose FROM #TABLES CHKBOX 2 +
TITLE .vTitle +
CAPTION .vCaption +
LINES 27 +
OPTION INITPOS 2,5 +
|WINDOW_BACK_COLOR WHITE +
|TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Tahoma +
|TITLE_FONT_COLOR NAVY +
|TITLE_FONT_SIZE 14 +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAME Tahoma +
|LIST_FONT_COLOR NAVY +
|LIST_FONT_SIZE 12 +
|BUTTONS_SHOW_Glyph ON +
|BUTTONS_BACK_COLOR WHITE
IF vChoose = '[Esc]' THEN
    GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
    CLEAR VARIABLES vTitle,vCaption,vChoose
    RETURN
```

```
-- Example Using INITVAL Option
```

```

IF (CVAL('DATABASE')) <> 'RRBYW20' OR (CVAL('DATABASE')) IS NULL THEN
  CONNECT RRBYW20 IDENTIFIED BY NONE
ENDIF
SET VAR vTitle TEXT = 'Choose Table(s)'
SET VAR vCaption TEXT = 'Using INITVAL Option in CHOOSE Command'
SET VAR vChoose TEXT = NULL
CLS
CHOOSE vChoose FROM #TABLES CHKBOX 3 +
TITLE .vTitle +
CAPTION .vCaption +
LINES 27 +
OPTION INITVAL Customer,InvoiceHeader,Titles +
|WINDOW_BACK_COLOR WHITE +
|TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Tahoma +
|TITLE_FONT_COLOR NAVY +
|TITLE_FONT_SIZE 14 +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAME Tahoma +
|LIST_FONT_COLOR NAVY +
|LIST_FONT_SIZE 12 +
|BUTTONS_SHOW_Glyph ON +
|BUTTONS_BACK_COLOR WHITE
IF vChoose = '[Esc]' THEN
  GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
  CLEAR VARIABLES vTitle,vCaption,vChoose
  RETURN

```

1.5.7.1.1 Title

These parameters are specific to changing how the CHOOSE title is displayed.

Parameter	Value	Description
TITLE_BACK_COLOR	value	Changes the background color of the title area. User can specify the integer value or use one of the predefined values.
TITLE_FONT_NAME	value	Specifies a font name for the title area. Values would consist of the fonts available on the computer.
TITLE_FONT_SIZE	value	Specifies a font size in title area.
TITLE_FONT_COLOR	value	Changes the font color of the title area. User can specify the integer value or type one of the predefined names.
TITLE_BOLD	ON OFF	Makes font in title area bold style.
TITLE_ITALIC	ON OFF	Changes the font in title area italic style.
TITLE_UNDERLINE	ON OFF	Changes the font in title area underlined.
TITLE_STRIKEOUT	ON OFF	Makes the font in title area strikeout.

1.5.7.1.2 List

These parameters are specific to changing how the CHOOSE list is displayed.

Parameter	Value	Description
VERTICAL_LINES	ON OFF	Specifies if vertical lines will be displayed. Use <code>VERTICAL_LINE_BREAKS</code> to define line locations.
VERTICAL_LINE_BREAKS	values	Specifies comma delimited pixel values for the location of the vertical lines (e.g. 150,185, 255)
TEXT_BREAKS	values	Specifies comma delimited character positions for the text breaks in the list to occur (e.g. 36, 112)
HEADERS	values	Specifies a comma delimited list of column headers (e.g. Table Name, Count, Description)
HORIZONTAL_LINES	ON OFF	Specifies if horizontal lines display between list options
LINE_COLOR	value	Specifies the horizontal line color
ZEBRA_STRIPES	ON OFF	Specifies if the zebra stripe colors are displayed
ZEBRA_ODD_COLOR	value	Specifies the zebra stripe color for odd number list items
ZEBRA_EVEN_COLOR	value	Specifies the zebra stripe color for even number list items
LIST_BACK_COLOR	value	Changes the background color of the list area. User can specify the integer value or type one of the predefined values.
LIST_FONT_NAME	value	Specifies a font name in the list area. Values would consist of the fonts available on the computer.
LIST_FONT_SIZE	value	Specifies a font size in the list area.
LIST_FONT_COLOR	value	Changes font color of the list area. User can specify the integer value or type one of the predefined values.
LIST_BOLD	ON OFF	Makes the font in the list area bold style.
LIST_ITALIC	ON OFF	Makes the font in the list area italic style.
LIST_UNDERLINE	ON OFF	Makes the font in the list area underlined.
LIST_STRIKEOUT	ON OFF	Makes the font in the list area strikeout

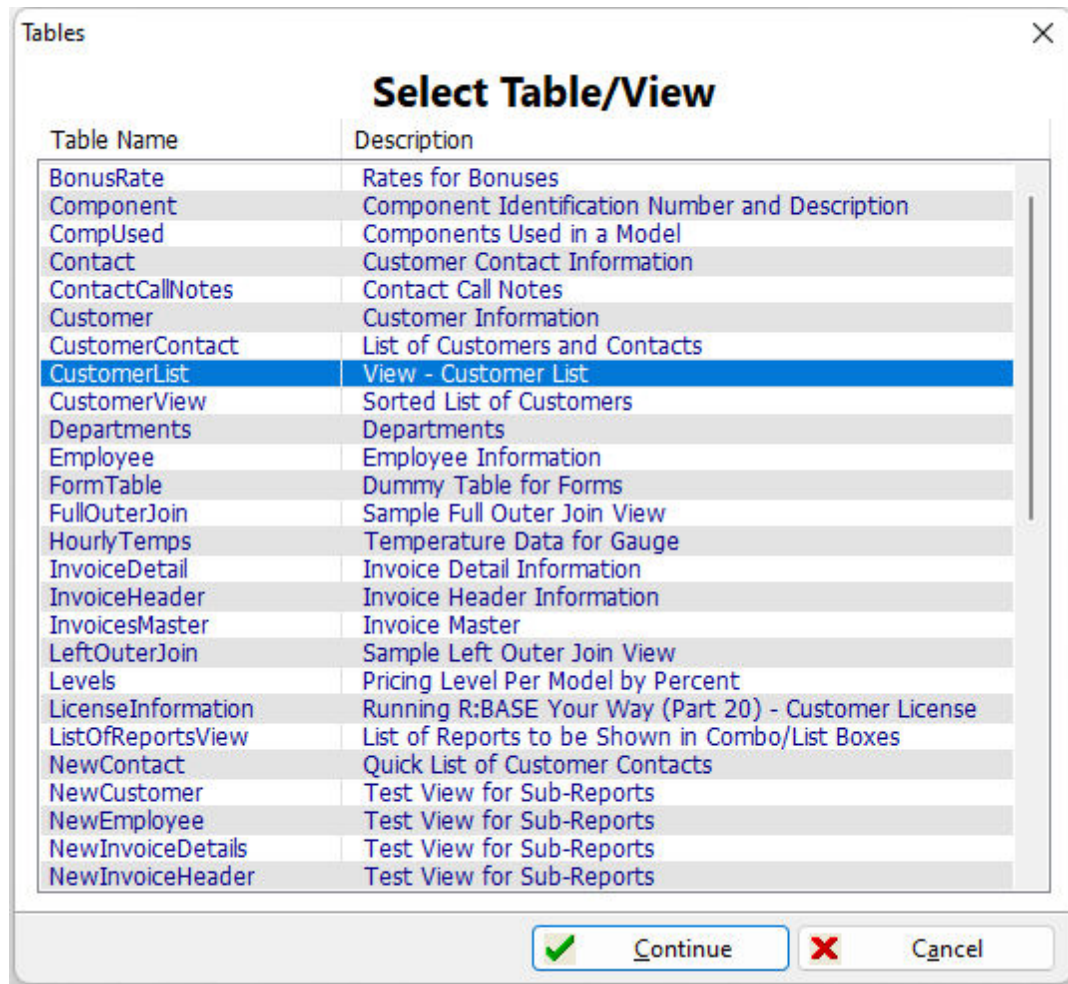
Example ("List View" Options):

```
CHOOSE vTableName FROM #VALUES FOR +
((LJS(SYS_TABLE_NAME,35))&LJS(SYS_COMMENT,75)),SYS_TABLE_NAME FROM SYS_TABLES +
WHERE SYS_TABLE_TYPE <> 'SYSTEM TABLE' ORDER BY SYS_TABLE_NAME ASC +
CHKBOX 1 TITLE 'Select Table/View' +
CAPTION 'Tables' LINES 26 FORMATTED +
OPTION TITLE_BACK_COLOR WHITE +
|TITLE_FONT_NAME Segoe UI +
|TITLE_FONT_SIZE 16 +
|TITLE_FONT_COLOR BLACK +
|TITLE_BOLD ON +
|LIST_BACK_COLOR WHITE +
|LIST_FONT_NAMETahoma +
|LIST_FONT_SIZE 9 +
|LIST_FONT_COLOR NAVY +
|WINDOW_BACK_COLOR WHITE +
|BUTTON_OK_CAPTION &Continue +
|BUTTON_CANCEL_CAPTION C&ancel +
|BUTTONS_SHOW_GLYPH ON +
|LINE_COLOR 3DLIGHT +
|ZEBRA_STRIPES ON +
```

```

|ZEBRA_ODD_COLOR 3DLIGHT +
|ZEBRA_EVEN_COLOR WINDOW +
|HORIZONTAL_LINES ON +
|VERTICAL_LINES ON +
|VERTICAL_LINE_BREAKS 150 +
|TEXT_BREAKS 36 +
|HEADERS Table Name,Description

```



1.5.7.1.3 Buttons

These parameters are specific to changing how the CHOOSE buttons are displayed.

Parameter	Value	Description
BUTTON_OK_CAPTION	value	Changes the caption for the "OK" button.
BUTTON_CANCEL_CAPTION	value	Changes the caption for the "Cancel" button.
BUTTONS_SHOW_GLYPH	ON OFF	Places images on the OK and Cancel buttons.
BUTTONS_BACK_COLOR	value	Changes the color of the button area.
BUTTONS_COLOR	value	Changes the color of the buttons.
BUTTONS_HEIGHT	value	Changes the height of the buttons.
BUTTONS_WIDTH	value	Changes the width of the buttons.
BUTTONS_FONT_NAME	value	Changes the font name for the text on the buttons.

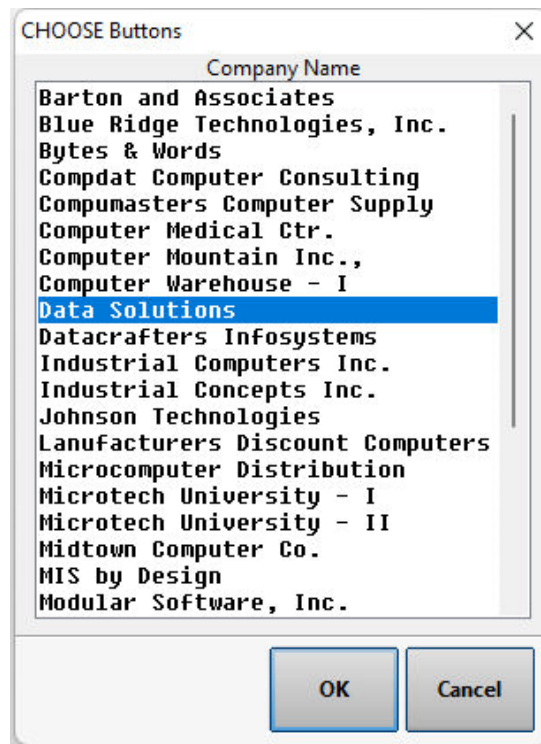
BUTTONS_FONT_BOLD	value	Specifies to use the bold style for the for the text on the buttons.
BUTTONS_FONT_ITALIC	ON OFF	Specifies to use the italic style for the for the text on the buttons.
BUTTONS_FONT_UNDERLINE	ON OFF	Specifies to use the underline style for the for the text on the buttons.
BUTTONS_FONT_STRIKEOUT	ON OFF	Specifies to use the strikeout style for the for the text on the buttons.

Example (Larger Button):

```

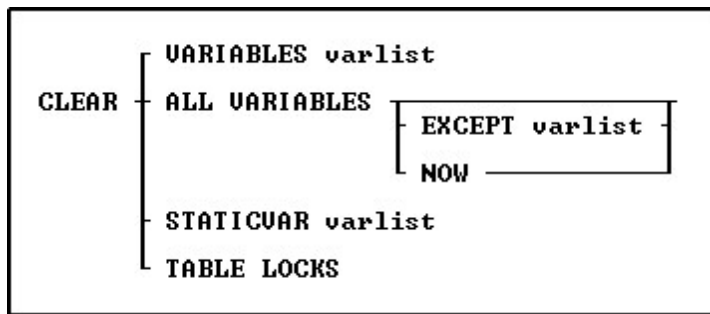
CHOOSE vCustIDTxt FROM #VALUES FOR +
Company,CustID FROM Customer ORDER BY Company +
CHKBOX 1 TITLE 'Company Name' +
CAPTION 'CHOOSE Buttons' LINES 20 FORMATTED +
OPTION BUTTONS_BACK_COLOR 16185078 +
|BUTTONS_COLOR SILVER +
|BUTTONS_HEIGHT 50+
|BUTTONS_FONT_BOLD ON

```



1.5.8 CLEAR

Use the CLEAR command to remove global variables from memory, static variables from a database, or clear table locks.



Options

ALL VARIABLES

Removes all global variables from memory.

EXCEPT varlist

Specifies variables that the CLEAR command will not remove. You can use wildcards in variable names.

NOW

Clears all the variable storage blocks (VSBs), including all printer control code variables and other non-permanent system variables (NPSVs). It re-allocates the blocks with just the permanent system variables (PSVs), #DATE, #TIME, #PI and SQLCODE. Without the fourth parameter, printer control code variables and other # variables are NOT cleared. Also, it does not completely re-initialize the memory blocks as does the CLEAR ALL VAR NOW. **The NOW argument is specific to DOS versions of R:BASE.**

STATICVAR varlist

Clears a list of one or more static variables. You can use wildcards in variable names. Static variables are created with the [SET STATICVAR](#) command.

TABLE LOCKS

Removes all locks on tables. This command parameter must be used with [MULTI](#) set to OFF and while connected to the database.

VARIABLES varlist

Removes a list of one or more variables. Use this option at the end of a complete set of procedures to clear variables that are no longer needed. You can use wildcards in variable names.

About the CLEAR Command

When R:BASE is first loaded into memory, only system variables are defined; they are not affected by the CLEAR command. Other variables you define remain in memory until you exit from R:BASE or use the CLEAR command.

Examples

The following command removes the global variables *vcounter* and *vname* from memory.

```
CLEAR VARIABLES vcounter, vname
```

The following command removes all global variables from memory.

```
CLEAR ALL VARIABLES
```

The following command removes all global variables except *var1*.

```
CLEAR ALL VARIABLES EXCEPT var1
```

The following command clears all variables beginning with the letter *v*.

```
CLEAR VARIABLES v%
```

The following command clears the vStartDate static variable.

```
CLEAR STATICVAR vStartDate
```

1.5.9 CLOSE

Use the CLOSE command to close an open cursor that was defined with the [DECLARE CURSOR](#) command.

CLOSE cursorname

Options

cursorname

Specifies a 1 to 18 character cursor name that has been previously specified by the DECLARE CURSOR command and opened with the [OPEN](#) command.

About the CLOSE Command

Cursors are pointers to rows in a table, and are defined using the DECLARE CURSOR command. When you no longer want to use the cursor but want to retain it for later use, use the CLOSE command to close the cursor. When you open the cursor again, it is positioned at the beginning of the set of rows defined by the DECLARE CURSOR command.

Use the [LIST CURSOR](#) command to list all of the currently defined cursors and whether the cursor is opened or closed.

When you close a cursor, most of the memory taken by the cursor definition is returned. If the DECLARE CURSOR command used any file handles, they are released.

Example

The following command makes the rows defined by the DECLARE CURSOR command as *cursor1* unavailable. To use the information defined by *cursor1* again, you need to reopen the cursor with the OPEN command.

```
CLOSE cursor1
```

1.5.10 CLOSEWINDOW

Use the CLOSEWINDOW command to close a form from within an application or entry/exit procedure (EEP). Without the alias it closes the current form.

CLOSEWINDOW [alias]

Options

alias

Specifies a window name, which is the name given to the instance of an MDI form started with the "AS *alias*" option when using the [ENTER](#), [EDIT USING](#), or [BROWSE USING](#) commands.

1.5.11 CLS

Use the CLS command to clear the R> Prompt window or [PAUSE](#) message box that is currently displayed.

```
CLS [ FROM scrnrow, scrncol [ TO scrnrow, scrncol ] [ backcolor ]
```

Options

backcolor

Specifies the background color of the cleared area. Valid R:BASE background colors are black, blue, brown, cyan, green, gray, magenta, and red. In Windows you can also use yellow and white.

FROM scrnrow, scrncol

Specifies the first screen row to clear, starting at the column specified by *scrncol*.

TO scrnrow, scrncol

Specifies the last screen row to clear, ending at the column specified by *scrncol*.

About the CLS Command

CLS does not issue a form feed. CLS replaces the [NEWPAGE](#) command in DOS to Windows conversions, which does issue a form feed. The additional CLS options pertain only to DOS versions of R:BASE.

Example

The following example clears the window from row 7 column 5 to row 9 column 60, and sets the background color in the cleared region to red.

```
CLS FROM 7,5 TO 9,60 RED
```

1.5.12 CODELOCK

Use the CODELOCK command to start CodeLock, the ASCII-to-binary conversion program that locks command and procedure files.

```
CODELOCK [ 1 cmdfile backfile binfile [ -p ]
          2 cmdfile apxfile procname
          3 scrfile apxfile screenname
          4 menufile apxfile menuname
          5 appfile apxfile
          6 inputspec outputfolder ]
```

Options

- 1**
Converts an ASCII command file to a binary command file.
- 2**
Adds an ASCII command file to a procedure file.
- 3**
Adds an ASCII screen file to a procedure file.
- 4**

Adds an ASCII menu file to a procedure file.

5

Converts an ASCII application file to a binary procedure file.

6

Converts a specified input folder location, and file group by extension, to binary procedure files within an output folder.

appfile

Specifies an ASCII application file.

apxfile

Specifies the name for the binary procedure file that is produced.

backfile

Specifies the name for the backup file.

binfile

Specifies the name for the binary command file that is produced.

cmdfile

Specifies an ASCII command file.

inputspec

Specifies the input folder location and file group by extension.

menufile

Specifies an ASCII menu file.

menuname

Specifies the name for the inserted menu block.

outputfolder

Specifies the output folder for the CodeLocked files.

-p

Forces CodeLock to use the older encryption method (used previous to R:BASE version 4.5 Plus!). This option is necessary if the application is run using R:BASE version 4.5 or earlier.

procname

Specifies the name for the inserted command block.

screenname

Specifies the name for the inserted screen block.

scrfile

Specifies an ASCII screen file.

About the CODELOCK Command

CodeLock is an ASCII-to-binary conversion program that encodes command and procedure files, and creates an .APX file from the source file. CodeLock creates files that can only be run-not viewed or edited.

The advantages to using files encoded by CodeLock are:

- The encoded file that R:BASE creates protects the source file from being altered because the binary files produced by CodeLock cannot be changed with a text editor.
- The run time for the encoded file might be reduced because the commands are already parsed or interpreted by CodeLock; some of the work the computer does to run the ASCII form of the command file is not necessary. Also, calls between binary blocks in procedure files can be done without opening and closing a different disk file for every call.

- A developer can create a complex application that can be distributed in one file instead of several stand-alone files.

Notes:

- The first eight (**8**) characters must be unique for Menu Blocks, Screen Blocks and Command Blocks in order to be uniquely identified.
- In R:BASE for Windows, when [MESSAGES](#) is set off, R:BASE does not display the file that is being encoded or any system messages while the file is being encoded. When MESSAGES is set on, the application file is displayed on screen as it is encoded.
- When you make additions or changes to an .APX file, the additions and changes are appended to the existing .APX file. If you make regular changes to an .APX file, you should make a new .APX file to avoid the .APX file from getting too large.
- Because CodeLock is only run from within R:BASE, R:BASE uses the current QUOTES setting while encoding an application. If R:BASE encounters a [SET QUOTES](#) command while encoding an application, R:BASE will use that QUOTES setting until it encounters another SET QUOTES command, or is finished encoding the application. When R:BASE finishes encoding the application, it resets the QUOTES setting to the setting that was in effect before CodeLock began.

Examples:

Example 01 (CodeLocks the ConComp.app file, creating ConComp.apx):

```
CODELOCK 5 ConComp.app ConComp.apx
```

Example 02 (CodeLocks all .rmd files within D:\CmpSource and places the binary files within D:\CmpLocked)

```
CODELOCK 6 D:\CmpSource\*.rmd D:\CmpLocked
```

1.5.13 COMMENT

Use comments in command or application files to provide internal program documentation.

There are a few comment designators: "--", "{ }", and "*(").

1. A "--" comment can be used only on a single line either by itself or following a command.

To comment an individual line, add two hyphen characters "--" to the beginning of the line. A carriage return at the end of the line indicates the end of the text for a comment that begins with two hyphens. In R:BASE Editor, the syntax highlighting will alter the display and change the font color to pink and the style to italicized. In the following example;

```
CLEAR VAR vResult
```

the command will become:

```
--CLEAR VAR vResult
```

2. A "{ }" comment may share a command line with a command, occupy a line itself, or extend over multiple command lines.

This designator is the recommended option with the latest releases of R:BASE, as it helps in avoiding any confusion when using parentheses with your R:BASE expressions. The set of squiggly brackets "{ }", with the desired commented text or commands enclosed within the squiggly brackets will comment the text. In the following example;

```
PLUGIN RPDFMerge 'vResult +
| ACTION MERGE +
| DOC_LIST_FILE PDFFilesToMerge.LST +
```

```
| SHOW_SETUP_DIALOG ON +
| OUTPUT_FILE OneBigMergedFile.PDF '
```

the command(s) will become:

```
{
  PLUGIN RPDFMerge 'vResult +
  | ACTION MERGE +
  | DOC_LIST_FILE PDFFilesToMerge.LST +
  | SHOW_SETUP_DIALOG ON +
  | OUTPUT_FILE OneBigMergedFile.PDF '
}
```

Keep in mind that any command(s) that are embedded within a multiple-line comment will not be executed.

3. A "*"() comment may share a command line with a command, occupy a line itself, or extend over multiple command lines.

Another use of characters that will comment your code is the asterisk character preceding a set of parentheses "*"()", with the desired commented text or commands enclosed within the parentheses. R:BASE interprets text following an asterisk and left parenthesis as a comment until a closing right parenthesis is reached. If the right parenthesis is not entered, R:BASE responds with a continuation prompt (+>). Enter a closing parentheses until you are returned to the R> Prompt, or other processing. In the following example;

```
PLUGIN RPDFMerge 'vResult +
| ACTION MERGE +
| DOC_LIST_FILE PDFFilesToMerge.LST +
| SHOW_SETUP_DIALOG ON +
| OUTPUT_FILE OneBigMergedFile.PDF '
```

the commands will become:

```
*(
  PLUGIN RPDFMerge 'vResult +
  | ACTION MERGE +
  | DOC_LIST_FILE PDFFilesToMerge.LST +
  | SHOW_SETUP_DIALOG ON +
  | OUTPUT_FILE OneBigMergedFile.PDF '
)
```

Although this option is still supported in R:BASE, it is now recommended that you use the squiggly bracket "{ }" method above. Keep in mind that any command(s) that is embedded within a multiple-line comment will not be executed.

Using Comment Designators

- Although used primarily in command files, you can enter a comment at the R> Prompt.
- If you place a comment on the same line as a command, leave at least one space between the comment and the command so the comment is not interpreted as part of the command.

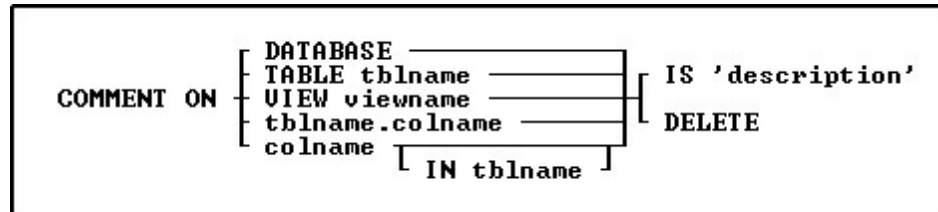
Restrictions on Using the Comments

- Do not include comments within the text of an ASCII menu file or a menu block because the comment will be read as part of the file.
- Do not embed comments within multi-line commands between continuation characters.

- Once starting a comment with a { bracket, everything except other left and right brackets is ignored until the matching right bracket.
- Once starting a comment with *(, everything except other left and right parentheses is ignored until the matching right parenthesis.

1.5.14 COMMENT ON

Use the COMMENT ON command to add a description to a database, table, view, or column.



Options

colname

Adds a description for a column in all tables in which it appears.

DATABASE

Adds a description for the connected database.

DELETE

Removes a description for a table or for a column in either the specified table or in all tables.

IN tblname

Adds a description for a column only in the specified table.

IS 'description'

Defines a description for a table or for a column in either the specified table or in all tables. The description is limited to 128 characters. The text must be enclosed in quotes using the current [QUOTES](#) setting.

TABLE tblname

Adds a description for the specified table.

VIEW viewname

Adds a description for the specified view.

tblname.colname

Specifies a column name. In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (tblname.colname).

About the COMMENT ON Command

Descriptions must be enclosed in single quotation marks ('), or the current delimiter character for QUOTES.

When you add a description to a column that appears in multiple tables, R:BASE adds the description to the column in every table. If you add a new table containing the column, you must add the description for that column to the new table.

Use the [LIST](#) command to list the descriptions of the tables and columns. R:BASE also displays the description when you modify a table using the Data Designer.

The database comment is displayed in the output of the LIST comment with the number of tables, columns, indexes, etc. The (CVAL('DBCOMMENT')) function can be used to retrieve the comment.

Comments are stored in the *SYS_COMMENTS* system table. When a column or table is renamed or removed, R:BASE automatically updates the *SYS_COMMENTS* table to reflect the change.

When access rights for a table have been assigned using the [GRANT](#) command, COMMENT ON requires the database owner's user identifier to describe tables and columns.

Examples

The following command adds a description to the *Employee* table.

```
COMMENT ON TABLE Employee IS 'Employee Information'
```

The following command adds a description to the *EmpID* column in all tables in the database.

```
COMMENT ON EmpID IS 'Employee Identification Number'
```

The following commands show two ways to add a description to the *EmpID* column in only the *Employee* table.

```
COMMENT ON Employee.EmpID IS 'Employee Identification Number'  
COMMENT ON EmpID IN Employee IS 'Employee Identification Number'
```

The following command removes the description from the *Employee* table.

```
COMMENT ON TABLE Employee DELETE
```

The following command removes the description from the *EmpID* column in every table in which the column occurs.

```
COMMENT ON EmpID DELETE
```

The following command removes the description from the *EmpID* column in the *Employee* table.

```
COMMENT ON EmpID IN Employee DELETE
```

The following command adds a description to the *UserManagement* database.

```
COMMENT ON DATABASE IS 'User and Contact Management System'
```

1.5.15 COMMIT

Use the COMMIT command to make permanent the changes to the data or database structure made since the beginning of the transaction. COMMIT is used during transaction processing.

COMMIT [WORK]

Options

WORK

WORK is optional and has no effect in R:BASE.

About the COMMIT Command

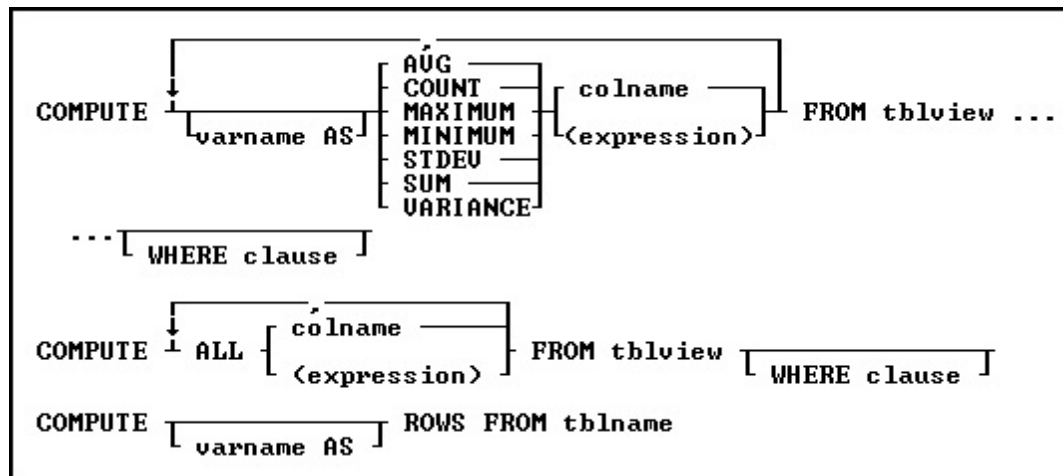
A transaction consists of all the commands executed since the last COMMIT or [ROLLBACK](#) command, or since you connected to the database if a COMMIT command has not been executed.

If you want to be able to process a group of commands, or transaction, without permanently affecting your data until you accept the changes, turn the [AUTOCOMMIT](#) setting off. R:BASE stores the information needed to reverse each modification made to the database in a Before Image file.

To accept a transaction and make the changes permanent, enter the COMMIT command after you have entered the transaction. The changes made by the commands in the transaction are made permanent and R:BASE clears the Before Image file to prepare it for the next transaction. The COMMIT command also releases all automatic table and database locks and executes any [SET LOCK OFF](#) commands included in the transaction. To reverse the transaction and undo the changes, enter the ROLLBACK command.

1.5.16 COMPUTE

Use the COMPUTE command to calculate expressions or perform mathematical operations on columns.



Options

' Indicates that this part of the command is repeatable.

ALL

Executes all of the functions relevant to the data type of the selected column.

AVG

Computes the numeric average. R:BASE rounds averages of INTEGER values to the nearest integer value and CURRENCY values to their nearest unit.

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (tblname.colname).

COUNT

Determines how many non-null entries there are for a particular column item.

(expression)

Determines the value for the column using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as #date, #time, and #pi.

FROM tblview

Specifies the table or view.

MAXIMUM

Selects the maximum numeric, time, date, or alphabetic value. For the TEXT data type, R:BASE evaluates the first 30 characters.

MINIMUM

Selects the minimum numeric, time, date, or alphabetic value. For the TEXT data type, R:BASE evaluates the first 30 characters.

ROWS FROM tblname

Counts rows in a specified table.

STDEV

Computes standard deviation. Standard deviation is a measure of how widely values are dispersed from the average value.

SUM

Computes the numeric sum.

VARIANCE

Determines variance.

varname AS

Specifies a global variable name in which to store the result of a COMPUTE command. If you use the arguments COUNT or ROWS, the variable must be an INTEGER data type.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the COMPUTE Command

R:BASE does not consider null values when it calculates averages, minimums, maximums, counts, sums, standard deviation, and variance.

The COMPUTE command can define a global variable with the data type that matches the results of the computation of a column's values.

You can also choose **Calculate** on the **Data Browser** menu in R:BASE for Windows to choose a column, then perform the mathematical operations that apply to the data type of the column.

It is recommended to use the [SELECT](#) Function(Column) notation over the COMPUTE notation as this provides greater compliance to the SQL standard and greater portability to new versions of R:BASE.

Examples

The following command computes, then stores the minimum on-hand quantity in the *vminvar* variable found in the *prodlocation* table.

```
COMPUTE vminvar AS MINIMUM onhand FROM prodlocation
```

The following command computes the total number of rows in the *transmaster* table and places the result in the *vrowcounter* variable.

```
COMPUTE vrowcounter AS ROWS FROM transmaster
```

The following command computes the average of an INTEGER data type (*onhand*) and returns the result as a DOUBLE data type.

```
COMPUTE AVG (onhand * 1.0) FROM prodlocation
```

The following command calculates your multiple computations on different columns. The command counts the number of entries in the *custid* column and averages the entries in the *netamount* column. The *custid* column and the *netamount* column are both from the *transmaster* table; all the specified columns must be in the same table or view.

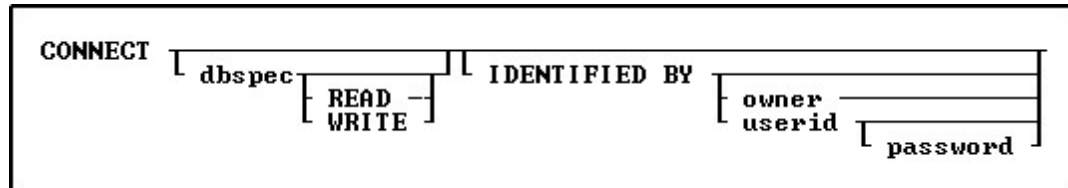
```
COMPUTE COUNT custid, AVG netamount FROM transmaster
```

The command below shows you how to use variables to hold the results of computations.

```
COMPUTE v1 AS COUNT custid,v2 AS AVG netamount FROM transmaster
```

1.5.17 CONNECT

Use the CONNECT command to open a database.



Options

dbspec

Opens a specified database.

IDENTIFIED BY

Specifies the user identifier. If left blank, R:BASE prompts you for the user identifier. R:BASE does not display it as you enter the text.

owner

Specifies the database owner name.

password

Optional; verifies the correct user identifier of the current user of the database.

READ

Opens a database as read-only. When you open a database as read-only, you cannot modify the data or database structure. All users must connect to the database in read mode if it is used.

userid

Specifies the user identifier. When access rights have been assigned using the [GRANT](#) command, you must enter an appropriate user identifier with the CONNECT command unless permission has been granted to PUBLIC.

WRITE

Opens a database in write/read mode. When you open a database in write/read mode, you can view and modify the database. This mode is the default when you use the CONNECT command without options.

About the CONNECT Command

R:BASE opens a database according to the current settings for the following SET commands: [FASTFK](#), [FASTLOCK](#), [MULTI](#), [ROWLOCKS](#), [TRANSACTIONS](#), and [STATICDB](#). Only one database can be open at a time.

You must open a database to use it; however, a database cannot be open when you copy, rename, or erase the database files.

When you use the CONNECT command without any of its options, R:BASE displays a list of the databases in the current directory. To open a database, highlight the database name and press [Enter]. This differs from older versions of R:BASE which required you to list a database in the connection command.

To open a database that is not in the current directory, add the drive and directory before you enter the database name, or use the [CHDIR](#) command before using CONNECT.

Database Access Rights with CONNECT

When you use the IDENTIFIED BY option, R:BASE identifies your level of access to the database. Only the last user identifier specified is in effect. If your user identifier contains spaces, enclose it in quotation marks when you enter it.

Closing a Database

R:BASE closes a database when you:

- Open another database.
- Issue a DISCONNECT command.
- Issue an EXIT command.
- Define a new database.

Updating Database Timestamps

When a database is created or opened, a timestamp is encoded in each of the four database files, enabling the database files to be stored in separate directories apart from one another. To ensure that the correct database files are accessed, R:BASE evaluates the timestamp values of the files as it searches for them on the path statement. R:BASE tries to connect to File 1 first, then it searches the path for the other database files.

Updating the encoded timestamp values in the four database files so that they all have the same timestamp values might be necessary if R:BASE is halted or if your computer or operating system crashes. When you try to connect to a database that is out of synchronization, R:BASE displays an error message. The [RBSYNC](#) command must be run.

Examples

The following command opens the *letters* database located on drive C: in the TOOLS/WP directory. R:BASE opens *letters* for reading and writing.

```
CONNECT c:\tools\wp\letters
```

The following command opens the *mydata* database with *user two* as the user identifier. The user identifier is enclosed in quotation marks because it contains a space.

```
CONNECT mydata IDENTIFIED BY 'user two'
```

The following example displays all databases in the current directory and allows you to select the database you wish to connect to.

```
CONNECT
```

1.5.18 CONTINUE

Use the CONTINUE command to move to the next occurrence of the [WHILE](#) loop and run the code.

CONTINUE

Example

In the following example, when the code is run, processing returns to line 3 after it completes the CONTINUE command on line 6. The while-block commands in line 8 are not run.

```
SET VARIABLE v1=0
SET VARIABLE v2=1
WHILE v1 = 0 THEN
  *(while-block commands)
  IF v2 <> 0 THEN
    CONTINUE
```

```
ENDIF
  *(while-block commands)
ENDWHILE
```

1.5.19 CONVERT

The CONVERT command is used to convert 4.5 and higher databases to R:BASE 11.

```
CONVERT dbname [ IDENTIFIED BY [ OWNER ] ]
```

Options

dbname

Specifies the database to be converted.

IDENTIFIED BY

Specifies the user identifier. If left blank, R:BASE prompts you for the user identifier. R:BASE does not display it as you enter the text.

OWNER

Optional; specifies the database owner name. If omitted and an OWNER name exists, you will be prompted.

About the CONVERT Command

R:BASE requires the conversion of legacy database formats. Once the database is converted, it CANNOT be accessed by any previous version of R:BASE. Be sure and backup your database before you convert it.

1.5.20 COPY

Use the COPY command to copy or combine files.

```
COPY filesource filetarget
COPY filesource1+filesource2 filetarget
```

Options

filesource

Specifies the file to be copied, the source file

filetarget

Specifies the file to receive the copy, the target file

About the COPY Command

This command is similar to the operating system COPY command.

When you copy files from one drive to another, include the drive letter of the drive you want to copy to.

When you copy files from one directory to another directory, the name of the directory you want to copy to must be different from the current directory.

The COPY command will combine multiple ASCII source files into a single target file.

Naming Copied Files

When you copy a file from one drive or directory to another drive or directory, you can keep the same name for the new file, or you can give the file a new name. If you want to make a copy of a file on the same directory, you must give the file a different name. If the new name is the name of an existing file, COPY replaces the existing file with the file you are copying.

Using Wildcards with the COPY Command

When copying a group of files, you can use the operating-system wildcards (?) or (*) in the file specification. The settings for the R:BASE special characters SINGLE and MANY do not affect the operating system wildcards.

When the target name is the same length as the source name, the command succeeds. However, if the target name is shorter or longer than the source name, the results are unpredictable.

Backing Up with the COPY Command

You can use the COPY command to backup your database, command files, and programs. When copying a database, copy all four database files as a set.

If you change the file extension when you use COPY to backup your database files, you must change the extension back to the original extension to use the database in R:BASE. The COPY command does not copy files onto multiple disks. If your database is too large to fit on one disk, use the [BACKUP](#) or [UNLOAD](#) commands.

Suggestion

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

Examples

The following command copies the file ThisFile.txt to ThatFile.txt on the current drive and directory.

```
COPY ThisFile.txt ThatFile.txt
```

The following command copies the MyData.dat file on drive C:\ to the current directory on drive E:| using the same file name.

```
COPY C:\MyData.dat E:\MyData.dat
```

The following command copies any files in the DBS directory that begin with MYDATA and have the extension .RX? to the ADMIN directory on the current drive using the same file names. The files with the .RX? extension could be the four disk files that contain an R:BASE database: MyData.RX1, MyData.RX2, MyData.RX3, and MyData.RX4.

```
COPY /DBS/MyData.rx? /admin
```

The following command copies the MYDATA database files using the new name SAVEDATA.

```
COPY MyData SaveData
```

The following command copies the MYDATA database files with an .RX? extension and changes the extension to .SV?

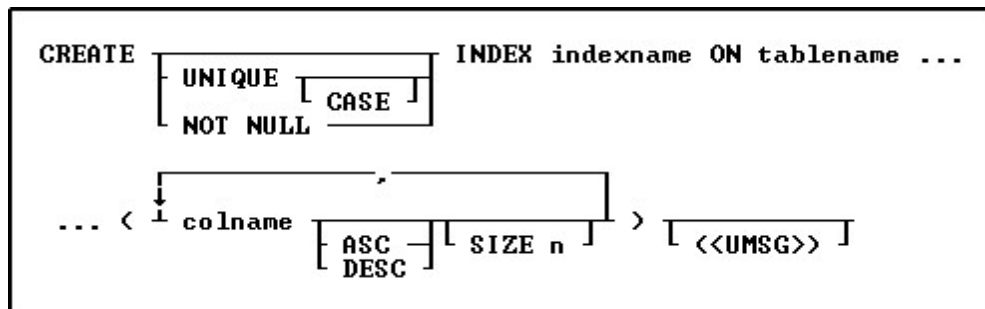
```
COPY MyData.rx? MyData.sv?
```

The following command combines the files FileHeader.txt and FileList.txt into FullContent.txt

```
COPY FileHeader.txt+FileList.txt FullContent.txt
```

1.5.21 CREATE INDEX

Use the CREATE INDEX command to speed up data retrieval by creating pointers that locate rows in a table easily.



Options

'
Indicates that this part of the command is repeatable.

ASC DESC

Specifies whether to sort a column in ascending or descending order.

CASE

Specifies that the data values will be case sensitive.

colname

Specifies a column name. The column name is limited to 128 characters. In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command.

INDEX indexname

Specifies an index, which is displayed with the [LIST INDEX](#) command. An *indexname* is required.

NOT NULL

Specifies a not NULL (partial) index is created. The index is populated with references to rows where the column value is not NULL. NULL values in the column are not added to the index. Not NULL indexes are ideal for columns which contain a considerable amount of NULL values, and where the not-NULL data is ideal for index retrievals. If a WHERE clause includes "ColName IS NULL" then the index is not used.

ON tblname

Specifies the table in which to create an index for a column.

SIZE n

Sets the minimum number of characters to preserve to determine uniqueness during hashing. This number can be a maximum of 196 characters. The index is created with the first *n* characters preserved and the rest of the value stored as a 4-byte hashed representation.

UNIQUE

Requires the values in a column to be unique.

(<UMSG>)

Creates a constraint violation message to appear whenever a unique index data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a unique number" for a unique index constraint violation. The index must be dropped, then recreated in order to modify the violation message.

About the CREATE INDEX Command

An index creates pointers to rows in columns, which allows R:BASE to find the rows using pointers much faster than searching the rows of data sequentially. You can index a column of any data type. An indexed column improves the performance of the following commands, clauses, and operations.

Commands, Clauses, and Operations to Use with Indexes	
DELETE DUPLICATES	RULES
INTERSECT	SELECT (when it includes a WHERE or an ORDER BY clause)
JOIN	SUBTRACT
Look-up expressions in Forms or Reports	UNION
ORDER BY	WHERE
PROJECT	

Although indexes speed up processing, they might slow down data entry because building an index for each value as it is entered takes time. Creating indexes for columns that contain many duplicate values does not always speed up processing. Indexes also occupy space on a disk.

Null Values

An indexed column can contain null values, but R:BASE uses an index most efficiently if each row in the indexed column contains a value. Primary keys, unique keys, or unique indexes explicitly restrict the insertion of null values. For other indexes, you can define a rule to ensure that a column always contains a value. A Not NULL index may also be defined for columns where many NULL values exist.

UPDATE Permission

When access rights for a table have been assigned using the [GRANT](#) command, you must have UPDATE permission for the column you want to index.

Indexing Criteria

Some columns are better candidates than others for indexing. To receive the greatest benefit from indexes, use the following criteria to help you decide which type of column is the best choice for indexing your table(s):

Primary Key

R:BASE automatically indexes the column(s) that is defined as the table's primary key.

Foreign Key

R:BASE automatically indexes the column(s) that is defined as the table's foreign key.

Columns Used in Queries

Columns that are not primary or foreign keys but are frequently used in queries should be indexed. Create a unique key constraint for columns that are not primary or foreign keys, but which uniquely identify a row in the table.

Columns Frequently Using ORDER BY or GROUP BY

Include a column in an ascending-order index when the column is not a primary or foreign key but is frequently referenced in an ascending-column ORDER BY or [GROUP BY](#) clause. Similarly, include a column in a descending-order index when the column is frequently referenced in a descending-column ORDER BY clause.

Full- and Partial-Text Indexes

Text columns can make effective indexed columns. If the size of the column that has a TEXT data type is 200 bytes or less, R:BASE creates a full-text index. A full-text index is an index that stores the entire contents of a column as an index in File 3, which is the file that contains indexes to columns. If the size of the column is greater than 200 bytes, R:BASE creates a partial-text index.

If you specify the SIZE option to be less than the defined length of a column, R:BASE creates a partial-text index, and any text column that has a defined length over 200 bytes must be a partial-text index. For columns that have a TEXT data type and exceed 200 bytes, you can specify the SIZE option to be between 0 and 196 to create a partial-text index. Specifying the size allows you to base your index on a specified number of characters at the beginning of the columns and to hash the remaining characters. For example, you can index a 225-character column with a TEXT data type by specifying the SIZE option to be any number less than 197 bytes. R:BASE will create an index with the first *n* characters and the rest of the value will be stored as a four-byte hashed representation of the text.

Partial-text indexes minimize storage space. However, partial-text indexes might not be as efficient as a full-text index, for example:

```
CREATE TABLE Cities (CityName TEXT(40), State TEXT(2), Country TEXT(20))
CREATE INDEX CityIndex ON Cities (CityName, State)
INSERT INTO Cities VALUES('Bellevue','WA','USA')
INSERT INTO Cities VALUES('Belltown','PA','USA')
SELECT CityName, State from Cities WHERE CityName = 'Bellevue'
```

In the above example, because the query reads data only from the index named *CityIndex*, there is no need to read the actual data stored in File 2—which is the data file—so the query is done quickly. The query is an index-only retrieval and produces fast results.

If a partial-text index was used in the same query as above, the partial-text index could also only use the index named *CityIndex*. Because the partial-text index only preserves the first four characters, it is impossible to return the correct answer to the query from the index. The query, as shown below, would slow processing because R:BASE must read data from the R:BASE data file.

```
CREATE INDEX CityIndex ON Cities (CityName SIZE 4, State)
SELECT CityName, State from Cities WHERE CityName = 'Bellevue'
```

When creating text indexes, be aware of the following:

- If you omit the **SIZE** option and the text field in the column is greater than 200 bytes, R:BASE creates a partial text index by storing the first 32 bytes of each field and hashing the remaining bytes in each field into a four-byte numeric representation of the text. For example, if the text is 280 bytes and you do not specify a size, R:BASE stores the first 32 bytes of each field and hashes the remaining 248 bytes into a four-byte integer.
- If you specify the **SIZE** option to be 16 bytes for a 60-byte column with a **TEXT** data type, R:BASE stores the first 16 bytes of each 60-byte text field and hashes the remaining bytes in each field into a four-byte numeric representation of the text. The total length of each index entry will be 20 bytes (16 + 4).
- If you specify the **SIZE** option to be 30 bytes for a 250-byte column with a **TEXT** data type, R:BASE stores the first 30 bytes of each 250-byte field and hashes the remaining bytes in each field into a four-byte numeric representation of the text. The total length of each index entry will be 34 bytes.
- If you specify the **SIZE** option to be 250 bytes for a column with a **TEXT** data type, you have made an illegal request because the maximum value for the **SIZE** option is 196 bytes when the length of the text field is greater than 200 bytes. If you specified the **SIZE** option to be 196 bytes for a 250-byte column, R:BASE would hash the remaining 54 bytes into a four-byte numeric representation of the text.
- If you omit the **SIZE** option and the text field in the column is 200 bytes or less, R:BASE creates a full-text index. For example, if the text is 80 bytes and you do not specify a size, R:BASE builds a full-text index of 80 bytes.

MICRORIM_INDEXLOCK

The system variable, **MICRORIM_INDEXLOCK**, is available to control concurrency locks for the **CREATE INDEX** command. This variable prevents **CREATE INDEX** from holding a permanent database lock. It locks only as necessary, allowing users access to the database. This results in longer index creation time but greater concurrency. **MICRORIM_INDEXLOCK** is set to any integer value.

Examples

The following command creates an index for the *CustID* column in the *TransMaster* table.

```
CREATE INDEX TranCust ON TransMaster (CustID)
```

The following example creates a multi-column index for the *Company*, *CustAddress*, and *CustState* columns in the *Customer* table.

```
CREATE INDEX CustAddr ON Customer (Company ASC, CustAddress ASC, CustState ASC)
```

The following example creates a unique index for the *TRStockID* column in the *TStockHeader* table.

```
CREATE UNIQUE INDEX TRSID ON `TStockHeader` (`TRStockID` ASC ) +  
( 'Values for rows in TurnRoundID must be unique!' )
```

The following creates a not NULL index for the *DiscontinuedDate* column in the *Products* table.

```
CREATE NOT NULL INDEX DiscontDate ON Products (DiscontinuedDate DESC)
```

1.5.22 CREATE SCHEMA

Use the CREATE SCHEMA command to name a database and assign a user identifier for the database owner.

```
CREATE SCHEMA AUTHORIZATION dbname [ ownername ]
```

Options

AUTHORIZATION dbname

Specifies the name of the database.

ownername

Allows you to assign a unique identifier for the owner of the database.

About the CREATE SCHEMA Command

A database name is limited to 128 characters. The database name must begin with a letter, and can contain letters, numbers, and the following symbols: number or pound sign (#), dollar sign (\$), underscore (_), or percent sign (%). A database name cannot contain blanks or have a file extension, and must follow the naming conventions for R:BASE and the operating system.

R:BASE creates four database files with extensions: .RX1, .RX2, .RX3, and .RX4. After you name a database, you need to use other commands to define the tables, views, rules, and access rights for the database.

Assigning A Database Owner's User Identifier

A database owner's user identifier must begin with a letter and can contain letters, numbers, and the following symbols: number or pound sign (#), dollar sign (\$), underscore (_), or percent sign (%). If user identifiers are assigned to users, the database owner's user identifier must be unique among all user identifiers in the database. A database owner's user identifier can be a maximum of 128 characters.

If you do not specify a user identifier, R:BASE assigns the default user identifier, PUBLIC. Until a user identifier is assigned, anyone can modify the database structure, read, enter, change, or delete data. When an owner's user identifier is assigned to a database, the database is accessible only by the owner. To give other users access rights to the database, use the [GRANT](#) command.

You do not have to assign a user identifier when the database is created. To assign a user identifier after the database has been created, use the [RENAME OWNER](#) command or the **Utilities: User Privileges...** menu option.

CREATE SCHEMA stores the owner's case folding and collating tables from the configuration file in the database. Be sure to keep a record of the owner's user identifier in a safe place away from your computer. If you lose the owner's identifier, you cannot search the database to find it.

Building a Database

You can use CREATE SCHEMA and other [CREATE](#) commands as an alternative to creating a new database using the menu options in R:BASE for Windows.

When you run the CREATE SCHEMA command, R:BASE closes the currently open database (if one exists), then defines and opens a new database.

Transaction Processing and the CREATE SCHEMA Command

If transaction processing is on when you enter a CREATE SCHEMA command, R:BASE first commits your current transaction (if any), then creates and connects you to the database. Transaction processing is on in the database, but you cannot reverse the CREATE SCHEMA command.

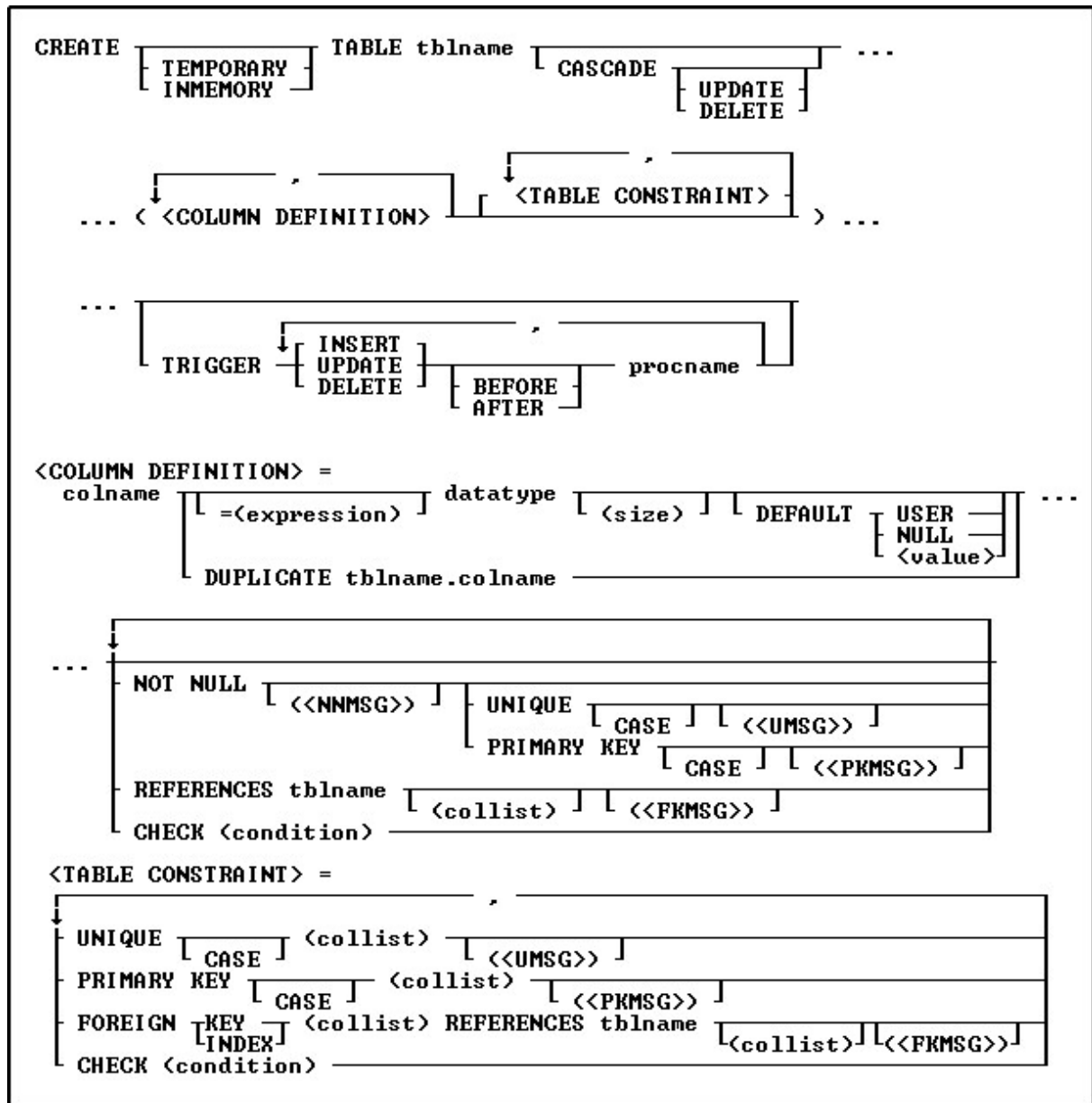
Example

The following command names the *finance* database and assigns *jane* as the database owner's user identifier.

```
CREATE SCHEMA AUTHORIZATION finance jane
```

1.5.23 CREATE TABLE

Use the CREATE TABLE command to define a new table in an existing database.



Options

'
Indicates that this part of the command is repeatable.

AFTER

Sets the trigger to activate after the INSERT, UPDATE or DELETE action.

BEFORE

Sets the trigger to activate before the INSERT, UPDATE or DELETE action. This is the default setting if the BEFORE/AFTER parameter is unused.

CASCADE

Maintains primary/foreign key relationships automatically. For example, if you either UPDATE or DELETE a primary key value from a table, the corresponding foreign key values are updated or deleted automatically. A CASCADE can be applied to UPDATE, DELETE or BOTH to specific primary keys. By not specifying either UPDATE or DELETE, both CASCADE restrictions will be enforced upon the primary/foreign key tables. Separate UPDATE and DELETE data restrictions can allow a CASCADE to be enforced for records that are updated, but not enforced when records are deleted, in order to avoid an accidental or undesired record delete. CASCADE can only be added to tables with primary keys.

CASE

Specifies that the data values will be case sensitive.

CHECK (condition)

Sets a condition to be satisfied before an update or insertion of a row can occur, which creates an R:BASE rule.

(collist)

Specifies a list of one or more column names, separated by a comma (or the current delimiter), used in the unique key specification. This option is only used when referencing a unique key.

colname

Specifies a column name. The column name is limited to 128 characters.

datatype

Specifies an R:BASE data type.

DEFAULT

Specifies a default value for the column if no value is provided by the user. The specified default value must match the column's data type (e.g. "PA" for a column that stores state address abbreviations). With TEXT columns, do not enclose the default value in the QUOTES character. The system variables #DATE, #TIME, and #NOW may be specified as default values. Static and global variables may also be specified as default values. When using system variables, static variables, or global variables, the value must be a dotted variable (the period must be included).

DUPLICATE tblname.colname

Copies the data type, length, default value, and NOT NULL flag if any. DUPLICATE does not support computed columns, and will not duplicate any comment/description for the column.

= (expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as #date, #time, and #pi.

(<FKMSG>)

Creates a constraint violation message to appear whenever a foreign-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a foreign-key constraint violation. You can define two messages: one for inserting and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

FOREIGN INDEX

With the FASTFK setting on, creates a foreign key that has an index using row pointers for data retrieval on selected columns.

FOREIGN KEY

Specifies a column or set of columns required to match values in a particular primary key or unique key defined in a table.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using INMEMORY in place of TEMPORARY creates the temporary table in memory, rather than a temporary file.

(<NNMSG>)

Creates a constraint violation message to appear whenever a not-null data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a not-null constraint violation. A constraint must be dropped, then recreated in order to modify the violation message.

NOT NULL

Prevents a column from accepting null values, but permits it to accept duplicate values. If this option is specified without a setting for a default value, you cannot insert rows without specifying values for the given column.

(<PKMSG>)

Creates a constraint violation message to appear whenever a primary-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a primary-key constraint violation. You can define three messages: one for uniqueness, one for deleting, and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

PRIMARY KEY

Specifies the column(s) to designate as a primary key.

procname

The procedure name. If a procedure by this name already exists in the database, an error is generated.

REFERENCES tablename

Identifies the primary key or unique key table to which the foreign key refers.

(size)

Defines the length of a column of the TEXT data type (if not the default 8). Defines the precision and scale of a column of the DECIMAL or NUMERIC data type, if not the default of precision 9 and scale 0 (9,0). VARBIT, VARCHAR, and BIT either require or can have a size.

tblname

Specifies a table name. The table name is limited to 128 characters.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

TRIGGER

Runs a stored procedure when an UPDATE, DELETE, or INSERT command is run on the table.

(<UMSG>)

Creates a constraint violation message to appear whenever a unique-key data integrity violation occurs. The message can suit the meaning of your data, such as "You must enter a valid number" for a unique-key constraint violation. You can define three messages: one for uniqueness, one for deleting, and one for updating. A constraint must be dropped, then recreated in order to modify the violation message.

UNIQUE

Requires the values in a column to be unique by defining a unique key constraint.

USER**NULL****(value)**

Default USER: Specifies the default value to be the user identifier.

Default NULL: Specifies the default value to be null.

Default (value): Specifies the default to be the indicated value.

About the CREATE TABLE Command

To define a table, you need to specify column definitions. Table and column names must begin with an upper- or lowercase letter. Names can contain letters, numbers, and the following special characters: #, \$, _, and %. R:BASE verifies that a table or column name is unique by reading all characters.

When you define a table, you can also add table constraints. However, you cannot add a foreign key to a temporary table.

To define more than one column in a command, use commas (or the current delimiter character) to separate the column definitions.

Computed Columns

A computed column is a column containing a value that R:BASE calculates from an expression defined for the column. In the expression, you can use other columns from the table, constant values, functions, and the system variables *#date*, *#time*, and *#pi*. Global variables are not allowed in an expression.

You must assign a data type that is compatible with the result of the computation. The columns used for calculating the computed column must precede the computed column in the table.

Indexing Columns

If you want to assign an index to a column, use the [CREATE INDEX](#) command.

Database Access Rights with CREATE TABLE

CREATE TABLE requires either the CREATE access right or the owner's user identifier when access rights have been assigned with the [GRANT](#) command.

Examples

The command below defines a table named *employee* with the following columns and data types: *empid* (INTEGER), *emptitle* (TEXT 30), *empfname* (TEXT 10), *emplname* (TEXT 16), *empaddress* (TEXT 30), *empcity* (TEXT 20), *empstate* (TEXT 2), *empzip* (TEXT 10), *empphone* (TEXT 12), *empext* (INTEGER), *hiredate* (DATE), and *entrydate* (DATE). In addition, the NOT NULL option specifies that the columns *empfname*, *emplname*, and *hiredate* must contain a value. The NOT NULL UNIQUE option specifies that the *empid* and *empext* columns must contain unique values.

```
CREATE TABLE employee (empid INTEGER NOT NULL UNIQUE, +
emptitle TEXT (30), empfname TEXT (10) NOT NULL, emplname TEXT +
(16) NOT NULL, empaddress TEXT (30), empcity TEXT (20), empstate +
TEXT (2), empzip TEXT (10), empphone TEXT (12), empext INTEGER +
NOT NULL UNIQUE, hiredate DATE NOT NULL, entrydate DATE)
```

The following command creates a table using the column constraint CHECK on the *empid* column.

```
CREATE TABLE employee (empid INTEGER CHECK (empid > 0), +
empname TEXT (40), empage INTEGER)
```

The example below creates a table using the column constraint CHECK on the *empid* and *empage* columns.

```
CREATE TABLE employee (empid INTEGER CHECK (empid > 0), +
empname TEXT (40), empage INTEGER CHECK (empage > 0 and empage < 100))
```

Table constraints are defined if it is necessary to reference multiple columns within the same expression. The UNIQUE (*collist*) option is entered at the end of the following command so that the values in the *empid*, *empfname*, *emplname* columns are unique as a group in a row. Because this option follows a column definition, precede the option with a comma.

```
CREATE TABLE employee (empid INTEGER NOT NULL, +
emptitle TEXT (30), empfname TEXT (10) NOT NULL, emplname TEXT +
(16) NOT NULL, empaddress TEXT (30), empcity TEXT (20), empstate +
TEXT (2), empzip TEXT (10), empphone TEXT (12), empext INTEGER +
NOT NULL UNIQUE, hiredate DATE NOT NULL, entrydate DATE, +
UNIQUE (empid, empfname, emplname))
```

The following command creates a table that would contain an employee's total years of employment. The command places a column constraint on the *empid* and *empage* columns, and a table constraint on the *yrshere* and *ysanywhere* columns. The value entered for *yrshere* must be less than or equal to the value entered for *ysanywhere*.

```
CREATE TABLE employee (empid INTEGER CHECK (empid > 0), +
empname TEXT (40), empage INTEGER CHECK +
(empage > 0 and empage < 100), yrshere INTEGER, +
ysanywhere INTEGER, CHECK (yrshere <= ysanywhere))
```

1.5.24 CREATE VIEW

Use the CREATE VIEW command to define a view that combines columns from existing tables or views.

```
CREATE [ TEMPORARY ] VIEW viewname [ <collist> ] ...
... AS SELECT clause [ WITH CHECK OPTION ]
```

Options

AS SELECT clause

Specifies the columns and rows to include in the view. As a rule anything that is acceptable in an ordinary select clause will work here. If you are planning on using expressions you should considering using the [SELECT AS](#) notation to give each column meaningful names.

(collist)

Specifies a list of one or more column names or aliases, separated by a comma (or the current delimiter). These names will be the column headings displayed in the result of a [SELECT](#) command or the Data Browser.

TEMPORARY

Creates a temporary view that disappears when the database is disconnected.

viewname

Specifies a view name.

WITH CHECK OPTION

Specifies that a row cannot be added or updated unless it meets the conditions included in the [WHERE](#) clause (which is part of the SELECT clause). R:BASE uses this option only on a view that can be updated.

About the CREATE VIEW Command

CREATE VIEW defines a view to store in the *sys_views* table. You can use a stored view whenever necessary. Unlike a table, stored views contain no data. R:BASE collects data for the view from the source tables or views when a command calling the view is run.

A view is the most efficient way to gather data from separate tables or views into one location. A view that can be updated allows you to enter, change, and delete data from the source table. The number of tables in a view is dependent on available memory.

You can define a view containing a maximum of 400 columns from as many tables or views as memory allows. However, a view is still limited to the 32,786 character row-size limit. You can use the SQL symbol asterisk (*) to include all columns from all tables or views, or you can specify the columns you want to include. You can combine these two methods to include all columns from one table or view and specify columns from another table or view. You must separate column, table, and view names with commas (or the current delimiter character).

Avoiding Multiple Occurrences of Columns

When you use only an asterisk (*) in the [SELECT](#) clause, the view will contain all columns from all tables or views. If the tables or views contain common columns, the view will contain multiple occurrences of those columns.

To avoid multiple occurrences of common columns, specify which columns to include in the view. For example, to include all columns from one table but only certain columns from another table, use an * for the first table, then list the column names to be included from the second table. You can specify the columns for a view as *t1.**, *t2.col2*, *t2.col3*, where *t1.** specifies all columns from table *t1* and *t2.col2*, *t2.col3* specifies two columns from table *t2*. Be sure that the list does not include the common columns contained in the second table. When you use a combination of * and column names, you must specify the table with which * is associated. However, you can omit the table or correlation name for the columns listed individually if those columns occur in only one table in the view.

Duplicate Rows

If the tables forming a view contain duplicate rows, either individually or in combination with other tables in the view, multiple duplicate rows will be displayed. Usually, the presence of duplicate rows in a view indicates a database design problem. Check your database structure for design flaws such as redundant data storage.

Linking Columns

When you build a view from two or more tables or views, define the relationship between the source tables and views by identifying linking columns in a WHERE clause. Linking columns are columns that contain the same values; their names can be the same or different. For example, the following WHERE clause specifies that a view displays only those rows where the values in *t1.col1* are equal to the values in the common column *t2.col1*.

```
WHERE t1.col1 = t2.col1
```

UNION Operator

Use views to create a subset of information. The conditions that link the source tables or views in the WHERE clause will limit the rows used in the view. However, you can create a view that uses all the rows from two or more tables by using the [UNION](#) operator. The UNION operator allows you to specify different selection criteria for each table in the view, as shown in the **Examples** below.

Updating Views

You can update the data for columns in a view when the view does not contain a UNION operator, and its [SELECT](#) clause meets the following requirements:

- The clause does not specify [DISTINCT](#).
- The clause does not include a [sub-SELECT](#) command in the WHERE clause.
- The clause does not include a [GROUP BY](#) or [HAVING](#) clause.
- The clause does not include an [ORDER BY](#) clause.

When you add, change, or delete rows by updating a single table view, you also modify the data in the source table. In multi-table views you cannot add, edit, or delete rows. Any additions or changes to data made through a view are subject to all the user-defined rules specified for the table when it was constructed. In addition, if you specify the WITH CHECK OPTION for the view, you can only add or modify rows that meet the conditions defined in the WHERE clause.

You can only use the [DELETE](#), [EDIT](#), [INSERT](#), [LOAD](#), and [UPDATE](#) commands with views that can be updated. If a view cannot be updated, you can use the view only to display data or as the basis for reports.

Other Commands to be Used with CREATE VIEW

Use the following commands with any view created with CREATE VIEW.

Available Commands for Views Created with CREATE VIEW			
BROWSE	EDIT USING*	PRINT	REVOKE
COMPUTE	FORMS	PROJECT	SELECT
CREATE VIEW	GRANT	QUERY	TALLY
CROSSTAB	LBLPRINT	RBLABELS	UPDATE**
DROP	LIST	RENAME	

EDIT*	LIST ACCESS	REPORTS	
-----------------------	-----------------------------	-------------------------	--

* These commands will display data when using a multi-table View (a View based on multiple tables), but the data will not be editable. A View with a GROUP BY parameter is also not editable.

** The UPDATE command will not update data when using a multi-table View (a View based on multiple tables), as the data is not editable. A View with a GROUP BY parameter is also not editable.

Views Compared with Look-up Tables

If the data used in a report is stored in more than one table, using a view is more efficient than a driving table and several look-up tables because it takes less time to print the report using a view. Using a view is more efficient because R:BASE gathers the data for a view before, rather than during, printing.

Database Access Rights with CREATE VIEW

The access rights that can be assigned with the [GRANT](#) command depend on whether or not the view can be updated. The ALL PRIVILEGES, DELETE, INSERT, SELECT, and UPDATE access rights can be granted on a view that can be updated. You must have ALL PRIVILEGES or SELECT access rights on a table or view to include it in a view.

If you have been assigned the SELECT access right and the WITH GRANT OPTION has been assigned on all the source tables or views used in a view, you can grant both SELECT and the WITH GRANT OPTION to other users.

If you are the database owner or you have the WITH GRANT OPTION on a view, you can also assign access rights on stored views.

Changing Views

You cannot change a view at the R> Prompt. To change a view from the R> Prompt, you must delete the view by using the [DROP](#) command, then define a new view. However, you can use the Query Builder to change a view.

Examples

The following command defines a view that can be updated and specifies a subset of columns "*custid*, *company*, *custaddress*, *custcity*, *custstate*, and *custzip*" from one table, *customer*. The column list must match the number of columns in the SELECT clause; the names, however, can be different. The WHERE clause restricts the rows to those with zip codes ranging from 40001 through 49999. The WITH CHECK OPTION specifies that only rows that meet the condition included in the WHERE clause can be added or changed in the database.

```
CREATE VIEW cust_addr (custid, custcompany, custaddress, custcity, custstate, +
custzip) AS SELECT custid, company, custaddress, +
custcity, custstate, custzip FROM customer WHERE custzip +
BETWEEN 40001 AND 49999 WITH CHECK OPTION
```

The following command defines a view that will display only those rows from the *customer* and *transmaster* tables that have matching values in the common column *custid*. Therefore, only the rows that have customers who have had a transaction will be included in the view. The command line ORDER BY *custid* tells R:BASE to sort the rows and display them by the customer identification number.

```
CREATE VIEW cust_trans AS SELECT t1.custid, company, netamount +
FROM customer t1, transmaster t2 WHERE t1.custid = t2.custid +
ORDER BY custid
```

The following command defines a view that will display only those rows from the *customer* table where the values in the *custid* column do not exist in the *transmaster* table. Therefore, only the rows that have customers who have not had a transaction will be included in the view.

```
CREATE VIEW cust_notrans AS SELECT custid, company FROM +
customer WHERE custid NOT IN (SELECT custid FROM transmaster) +
ORDER BY custid
```

The following command combines the commands in the two preceding examples, creating a view that will display all rows from both the *customer* and the *transmaster* tables. The UNION operator joins the two SELECT clauses, allowing you to display rows for all customers whether or not they have had a transaction.

The first SELECT clause instructs R:BASE to include the rows from both tables where the values in *custid* match. The second SELECT clause instructs R:BASE to include rows from the *customer* table where there are no values for *custid* in the *transmaster* table.

When you use the UNION operator, the number of columns specified in the SELECT clauses must be the same and the data types of the columns must be compatible. If there is no column in one table that matches a column listed in the other table's SELECT clause, you must substitute a value (or null value). Because the *netamount* column does not exist in the *customer* table used in the second SELECT statement, the value \$0.00 was entered in place of *netamount*.

```
CREATE VIEW all_cust_trans AS SELECT t1.custid, company, netamount +
FROM customer t1, transmaster t2 WHERE t1.custid = t2.custid +
UNION SELECT custid, company, $0.00 FROM customer +
WHERE custid NOT IN (SELECT custid FROM transmaster) +
ORDER BY custid
```

1.5.25 CREATEOBJECT

Use the CREATEOBJECT command to create dynamic form objects on demand.

```
CREATEOBJECT <ControlType> <ComponentID> <ParentControl> <Left Top Width Height>
```

Options

ComponentID

The unique identifier that you assign to the component when you insert the object.

Height

The Height coordinate for the control

Left

The Left coordinate for the control, in relation to the left side of its parent

ControlType

The form control, or object type. The actual name of the Control Type that must be used in the CREATEOBJECT syntax are available from the Data Dictionary, and are listed as follows by Form Control category:

Standard Controls

- StaticText
- Button
- EnhancedPanel
- EnhancedGroupBox

Database Controls

- DBText
- DBEdit
- DBMemo
- DBWebBrowser
- DBMailLabel
- DBImage
- DBCheckBox
- DBRadioButton
- DBRadioGroup

DBLookUpComboBox
DBLookUpListBox
DBSpinEdit
DBUserComboBox
DBUserListBox
DBTreeView
DBLookUpListView
ScrollingRegion
EnhancedDBNavigator
EnhancedDBGrid
DBSlider
AdvancedDBRichText
DBTimePicker
EnhancedDBCalendar
DBSpinner
DBTreeGrid
DBTaskTracker
DBDateAndTimeEdit
DBOrgChart
DBGauge
DBTreeList
DBCARDView
DBVerticalGrid
DBPivotGrid

Variable Controls

VarLabel
VarEdit
VarMemo
VarWebBrowser
VarEmailLabel
VarImage
VarCheckBox
VarRadioButton
VarRadioButtonGroup
VarLookUpComboBox
VarLookUpListBox
VarSpinEdit
VarUserComboBox
VarUserListBox
VarLookUpListView
SystemVariable
VarTimePicker
EnhancedVarCalendar
VarSpinner
VarAdvancedRichEdit
VarGauge
VarDateTimePicker

Additional Controls

BitBtn
SpeedButton
DropDownButton
Image
Bevel
LedDisplay
TimeLedLabel
TreeView
ListView
GroupBar
Group
EnhancedTabControl
EnhancedTabSheet
EnhancedSpeedButton
StatusBar

ClockStatus
KeyStatus
StatusPane
GlyphStatus
MarqueeStatus
DBStateStatus
DBStatus
CurrentTablePane
CurrentFieldPane

PDFViewer
TabSet
LinkLabel
TimePicker
EnhancedCalendar
CollapsePanel
CountDownTimer
LayoutManager
SplitView
TileMenu

Internet Controls

WebBrowser
MailLabel

File System Controls

DirTree
FileBox

Legacy Controls

VerticalLine
HorizontalLine
3DBox
Panel
GroupBox
Wallpaper
EnhancedWallpaper
UnicodeLabel
OfficeButton
GIFImage
Shape
Animate
Splitter
TabControl
TabSheet
Clock
ShapeButton
Meter
Separator
Line
DBGrid
DBNavigator
DBRichEdit
DBDateTimePicker
DBCcalendar
DB Unicode Memo
VarDateTimePicker
VarCalendar
VarUnicodeMemo
VarRichEdit

ParentControl

The form object in which other controls are contained within. A parent control can be an actual form or any of the following form controls: Panel, Enhanced Panel, Group Box, Enhanced Group Box, Tab Control, Enhanced Tab Control, Status Bar, DB Radio Button Group, Variable Radio Button Group, Scrolling Region, etc.

Top

The Top coordinate for the control, in relation to the top of its parent

Width

The Width coordinate for the control

About the CREATEOBJECT Command

Any form object can be created at runtime, simply by knowing the exact Control Type, Component ID, parent (form or any control having the option of a container, and coordinates (Left, Top, Width, Height).

To place Database Control objects, you'll need to provide additional parameters, such as, Table Name, Field Name. For Variable Objects, you'll need to provide associated variable name. And, once the object is defined, it is required to initialize the object in order for the object to be placed on the form. All of the definitions for the newly created object are performed using the [PROPERTY](#) Command.

Use the Data Dictionary to browse the list of form Control Type and to easily insert the values into your command files/applications.

Order of Processing To Create an Object

The order of processing for the CREATEOBJECT command is as follows:

1. Use the CREATEOBJECT command to create the object.
2. For Database or Variable controls, use the [PROPERTY](#) command to assign the database field or Variable name.
3. When inserting Custom EEPs, use the PROPERTY command to assign the Custom EEP command syntax.
4. Use the **PROPERTY ComponentID INIT 'TRUE'** command to initialize the object.
5. Use the PROPERTY command to set any additional characteristics, like color, font, background, etc., that would be specific to the type of object you are creating.

Notes:

- A form must include the associated Master table, unless a variable form is being created.
- Any additional form related tables MUST be defined using the form designer.
- To delete an object on the fly, use the [DELETEOBJECT](#) command.

Examples

```
-- Place a text caption for the field (ObjectType: StaticText)
CREATEOBJECT StaticText 'StaticText1' 'RBASE_FORM' 10 10 400 20
PROPERTY StaticText1 INIT TRUE
PROPERTY StaticText1 CAPTION 'Dynamically Created DB Controls'

-- Place a Panel on the Form (ObjectType: Panel)
CREATEOBJECT Panel 'Panel' 'RBASE_FORM' 10 40 400 120
PROPERTY Panel INIT TRUE

-- DB Edit Control Based on Table: Customer, Column: CustAddress
CREATEOBJECT DBEdit 'DBEdit2' 'Panel' 100 35 220 20
PROPERTY DBEdit2 TblName 'Customer'
PROPERTY DBEdit2 FldName 'CustAddress'
PROPERTY DBEdit2 INIT TRUE
PROPERTY DBEdit2 FRAMEVISIBLE TRUE

-- Place a DB Navigator Control Based on Table: Customer
CREATEOBJECT DBNavigator 'Navigator' 'Panel' 100 90 200 20
PROPERTY Navigator TblName 'Customer'
PROPERTY Navigator INIT TRUE
PROPERTY Navigator IMAGESTYLE 'Razzmatazz'
```

```

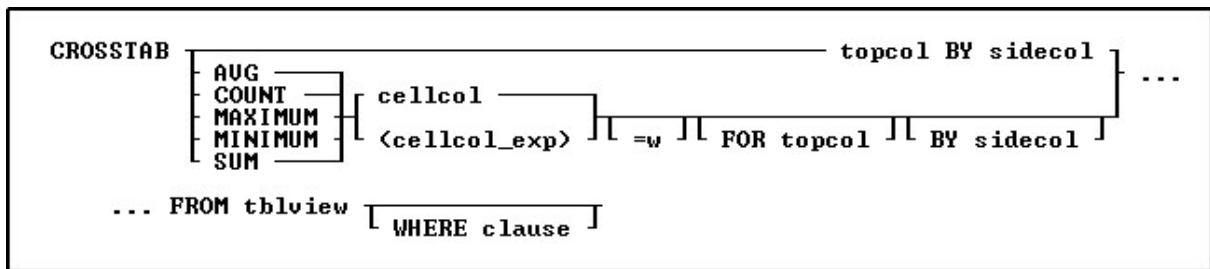
PROPERTY Navigator FLAT TRUE

-- To insert Custom EEP command syntax into the "DBEdit2" Control defined above
-- PROPERTY ComponentID 'Custom_EEP->YourCustomEEP' 'YourCustomEEPTText'
CLEAR VAR vCommand1
SET VAR vQuote = (CVAL('QUOTES'))
SET VAR vCommand1 TEXT = ('PAUSE 2 USING'&vQuote&'This is the
Address'&vQuote&';RETURN')
PROPERTY DBEdit2 'Custom_EEP->OnClickEEP' .vCommand1
RETURN

```

1.5.26 CROSSTAB

Use the CROSSTAB command to count the occurrences of each unique pair of values from two columns in a table or view.



Options

AVG

Computes the numeric average. R:BASE rounds averages of INTEGER values to the nearest integer value and CURRENCY values to their nearest unit.

BY sidecol

Specifies the column whose values are displayed down the side of the crosstab.

cellcol

Specifies the column whose values are computed and displayed in the cells portion of the crosstab.

COUNT

Determines how many non-null entries there are for a particular column item.

(cellcol_exp)

Specifies the column whose values are computed and displayed in the cells portion of the crosstab with an expression. The expression can include columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

FOR topcol

Specifies the column whose values are displayed across the top of the crosstab. If the *topcol* option is omitted, the *sidecol* and *cellcol* total or computation is displayed.

FROM tblview

Specifies the table or view.

MAXIMUM

Selects the maximum numeric, time, date, or alphabetic value in a column. For the TEXT data type, R:BASE evaluates the first 8 characters. For the NOTE data type, R:BASE evaluates the first 20 characters, but this permits text expressions from views to be used.

MINIMUM

Selects the minimum numeric, time, date, or alphabetic value in a column. For the TEXT data type, R:BASE evaluates the first 8 characters. For the NOTE data type, R:BASE evaluates the first 20 characters, but this permits text expressions from views to be used.

SUM

Computes the numeric sum.

topcol

Specifies the column whose values are displayed across the top of the crosstab.

=w

Sets the width, in number of characters, of the column displaying the results for *cellcol*.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the CROSSTAB Command

You can use CROSSTAB from the R> Prompt from the **Calculate** menu in the "Data Browser" window in R:BASE for Windows. R:BASE sorts the information in ascending order and displays it in tabular form. You can also display the results of numeric calculations performed on the values in the two columns in a third column. The CROSSTAB command does not consider null values in its calculations.

Examples

Example 01:

The following command shows the net sum of the transactions for the sales representative whose employee identification is 102, ordered by transaction date.

```
CROSSTAB SUM NetAmount FOR EmpID BY TransDate FROM TransMaster WHERE EmpID = 102
```

The results look similar to the following:

TransDate	102 (Total)	102
-----	-----	-----
03/15/2010	\$19,965.00	\$19,965.00
-----	-----	-----
	\$19,965.00	\$19,965.00

Example 02:

The following command shows the average of the transactions for the customer whose identification number is 117.

```
CROSSTAB AVG NetAmount FOR CustID FROM TransMaster WHERE CustID = 117
```

CustID	(Total)
-----	-----
117	\$6,337.50
-----	-----
	\$6,337.50

1.6 D

1.6.1 DBCONN

Use the DBCONN command to view all applications with connections to the currently connected database.

DBCONN

The list of applications may include R:BASE, Runtime for R:BASE, compiled applications, programs that have made ODBC connections to the databases, etc.

The output shows the application name, path, and file name.

```
R> DBCONN
```

```
Current connected database applications:  
R:BASE 11 (C:\RBTI\RBG11\RBG11.exe)
```

1.6.2 DEBUG

Precede any R:BASE command with the DEBUG command to control the running of the command. In forms runtime, [Ctrl] + [Shift] + [F11] will toggle the DEBUG flag.

DEBUG command

Option

command

Specifies a valid command name.

About the DEBUG Command

If [DEBUG](#) is set on, the DEBUG command runs; if it is set off, the DEBUG command is ignored. By allowing DEBUG to be set on or off, DEBUG statements in command files or applications can be activated or deactivated.

Examples

The following command lines show all variables and issue a [LIST](#) command.

```
SET DEBUG ON  
DEBUG SHOW VAR  
LIST  
SET DEBUG OFF
```

Because DEBUG is set off, the following command sequence only issues the LIST command.

```
SET DEBUG OFF  
DEBUG SHOW VAR  
LIST
```

1.6.3 DECLARE CURSOR

Use the DECLARE CURSOR command to create a cursor that points to a row in a table or view.

DECLARE cursorname [SCROLL] CURSOR FOR SELECT clause

Options

cursorname

Specifies a 1 to 18 character cursor name.

CURSOR FOR SELECT clause

Specifies the columns and rows from the table whose values you want to use. You may include the [DISTINCT](#) modifier as well as [WHERE](#) clauses and [ORDER BY](#) clauses.

SCROLL

Defines a cursor that moves forwards and backwards through a table. If this option is omitted, the cursor can only move forward.

About the DECLARE CURSOR Command

In the [SELECT](#) clause, specify the columns that contain the values you want to use from the row. Specifying the columns makes the column values accessible to the [FETCH](#) and [SET VARIABLE](#) commands. Once a cursor is declared, use the [OPEN](#) command to initialize the cursor and position it before the first row specified by DECLARE CURSOR.

Use DECLARE CURSOR to define a path through a table or view. You can move through the defined rows using the FETCH command by using either multiple FETCH commands or embedded FETCH commands within a WHILE loop. You only need to point to specific columns with DECLARE CURSOR, then FETCH can retrieve those columns by placing their values into variables. You can define a scrollable cursor, which is a cursor that moves backwards and forwards through a table.

DECLARE CURSOR defines a temporary view in memory; R:BASE does not store the view definition in the `sys_views` table. The SELECT clause defines columns, tables, rows, sort order, and potential grouping for the rows. When DECLARE CURSOR executes, it validates the syntax and names of columns and tables. The [OPEN](#) command can evaluate variables, create a copy of the cursor based on those values, then position the cursor before the first row.

Listing Cursors

Use [LIST CURSOR](#) to list all currently defined cursors and their status, open or closed.

Using Cursor Names in Commands

You can use the cursor name instead of a table name in commands. The following table provides examples of using the cursor name instead of a table name in commands.

To do this...	Use the cursor name like this...
Set a variable to a column value	FETCH <i>cursorname</i> INTO <i>varlist</i> SET VARIABLE <i>varname</i> = <i>colname</i> + WHERE CURRENT OF <i>cursorname</i>
Change a column value to a constant	UPDATE <i>tblname</i> SET <i>colname</i> = <i>value</i> + WHERE CURRENT OF <i>cursorname</i>
Change a column value to a variable value	UPDATE <i>tblname</i> SET <i>colname</i> = <i>.varname</i> + WHERE CURRENT OF <i>cursorname</i>
Change a column value to an expression	UPDATE <i>tblname</i> SET <i>colname</i> = (expression) + WHERE CURRENT OF <i>cursorname</i>
Delete the pointed-to row	DELETE FROM <i>tblname</i> + WHERE CURRENT OF <i>cursorname</i>

Modifying Data Using a Cursor

If you use a cursor in commands that modify data (the [UPDATE](#) and [DELETE](#) commands), only the current row is modified. To modify all referenced rows, include [FETCH](#) in a [WHILE](#) loop to move the cursor through the rows.

Checking for End-of-Data Conditions

End-of-data conditions determine whether you have reached the end of the data declared with the DECLARE CURSOR command. The three ways to check for end-of-data conditions are:

- Use an error variable defined with the [SET ERROR VARIABLE](#) command
- Use the *sqlcode* variable
- Include a [WHENEVER NOT FOUND](#) command

Closing Cursors

The following commands close cursors.

Command Name	Description
CLOSE	Closes the open cursor but does not remove the cursor definition. However using CLOSE frees most of the memory used when a cursor is opened. CLOSE also frees any file handles used by DECLARE CURSOR.
COMMIT	Closes any open cursors
CONNECT	Removes any cursor definitions from memory
DISCONNECT	Removes any cursor definitions from memory
DROP CURSOR	Entirely removes the cursor definition. Dropping a cursor definition frees all memory used by the definition.
ROLLBACK	Closes any open cursors

Examples

The following example uses the SCROLL option with DECLARE CURSOR.

```
DECLARE c1 SCROLL CURSOR FOR SELECT empid, transid, transdate, custid, netamount FROM
transmaster
```

Checking End-of-Data Conditions Using sqlcode

The following example uses *sqlcode* to check end-of-data conditions, which is the recommended program structure for DECLARE CURSOR. The *sqlcode* system variable holds values only for specific types of status.

Type of Error	SQLCODE
Data found	0
Data not found	100

In the following example, the [WHILE](#) statement checks the value of *sqlcode*.

```
1) DECLARE cursor1 CURSOR FOR SELECT custid, netamount +
    FROM transmaster ORDER BY netamount
2) OPEN cursor1
3) FETCH cursor1 INTO vcustid INDICATOR vi1, vnetamt +
    INDICATOR vi2
4) WHILE sqlcode <>100 THEN
    SHOW VARIABLE vcustid
    SHOW VARIABLE vnetamt
5)  FETCH cursor1 INTO vcustid INDICATOR vi1, vnetamt +
    INDICATOR vi2
    ENDWHILE
6) DROP CURSOR cursor1
```

1. DECLARE CURSOR defines the cursor path.
2. [OPEN](#) opens the cursor, evaluates variables, and positions the cursor before the first row.
3. The first [FETCH](#) command retrieves the first set of values. The indicator variables *vi1* and *vi2* capture the status values, -1 for null and 0 for a value. If you omit indicator variables in FETCH commands, R:BASE displays a message if it encounters a null value, but continues processing rows.
4. The [WHILE](#) loop processes the rows until there are no more rows. At that point, *sqlcode* is set to 100, and the WHILE loop ends. Control passes to the command after ENDWHILE. If the first FETCH retrieved no data, the WHILE loop is not entered.
5. FETCH retrieves all succeeding rows and sets *sqlcode* each time. When it does not find any more data, *sqlcode* is set to 100 and the WHILE loop ends.
6. [DROP CURSOR](#) removes the cursor definition from memory.

Using the WHENEVER Command with DECLARE CURSOR

The following example shows the use of the [WHENEVER](#) command, which checks the value of *sqlcode*. A single WHENEVER command can start a status-checking cycle that remains in operation until a command

or procedure file finishes running. As in the first two examples, an indicator variable is included with each variable in [FETCH](#). Without the indicator variables, R:BASE displays a message if it encounters a null value, but continues processing rows.

```

1) WHENEVER NOT FOUND GOTO skiploop
2) DECLARE cursor1 CURSOR FOR SELECT custid, netamount +
   FROM transmaster ORDER BY netamount
3) OPEN cursor1
4) FETCH cursor1 INTO vcustid INDICATOR vi1, vnetamt +
   INDICATOR vi2
5) WHILE #DATE IS NOT NULL THEN
   SHOW VARIABLE vcustid
   SHOW VARIABLE vnetamt
   FETCH cursor1 INTO vcustid INDICATOR vi1, vnetamt +
   INDICATOR vi2
   ENDWHILE
6) LABEL skiploop
7) DROP CURSOR cursor1

```

1. WHENEVER NOT FOUND tells R:BASE to execute [GOTO](#) if a command that searches for data, such as [FETCH](#), cannot find more rows. If the first [FETCH](#) command does not find any rows, control passes to the command following [LABEL skiploop](#). WHENEVER automatically checks any command that searches for data. If a data-not-found condition occurs, control passes to the command following the specified label.
2. DECLARE CURSOR defines the cursor path.
3. [OPEN](#) opens the cursor, evaluates the variables, and positions the cursor before the first row.
4. The first [FETCH](#) command retrieves the first set of values. If no rows match, control passes to [LABEL skiploop](#). Indicator variables *vi1* and *vi2* capture the status values (-1 for null and 0 for a value). If you omit indicator variables in [FETCH](#) commands, R:BASE displays a message if it encounters a null value, but continues processing rows. (WHENEVER instructs R:BASE to exit the [WHILE](#) loop only when *sqlcode* is 100.)
5. The WHILE loop processes rows until WHENEVER stops execution.
6. This label defines where to pass control if a data-not-found condition occurs before the WHILE loop begins executing. WHENEVER includes this label name.
7. [DROP CURSOR](#) removes the cursor definition from memory.

Visit the [From The Edge](#) Web site to download the "R:BASE Cursors Explained" technical document.

1.6.4 DEFF

Use the DEFF command to open the External Form Designer, or to open the designer with a specific file.

```

DEFF [ filename.rff ] [ MODAL ]

```

filename.rff

Specifies the name of the external form file.

MODAL

Launches an external form in the designer from a command block, command file, or EEP. The External Form Designer must be closed in order to continue the next command. Specifying a new file name will create a new external form.

Example

The following command launches the *TileMenu_Sample* external form within the External Form Designer.

DEFF TileMenu_Sample.rff MODAL

1.6.5 DELETE

Use the DELETE command to remove selected rows from a table.

```
DELETE [ ROWS ] FROM tblview [ WHERE clause
                                WHERE CURRENT OF cursor ]
DELETE filespec
```

Options

FROM tblview

Specifies the table or view.

ROWS

This word is optional.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

WHERE CURRENT OF cursor

Specifies a cursor pointing to the row the [DELETE](#) command will remove. This option can replace a standard WHERE clause.

Use the [DECLARE CURSOR](#) command to define the cursor.

filespec

Specifies the file to be deleted. Optionally, include a drive and path specification in the form D:\PATHNAME\FILENAME.EXT.

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

About the DELETE Command

DELETE removes rows from a table or view. Without a WHERE or WHERE CURRENT OF clause, R:BASE deletes all rows from the specified table or view. R:BASE displays a confirmation message before deleting the rows. R:BASE does not display a confirmation message when you execute a DELETE command from a command file. Views must be updatable to delete rows from it; for more information about updatable views, see [CREATE VIEW](#).

Before you use a WHERE clause with the DELETE command, test the clause by using it with a [SELECT](#) command, which allows you to view the rows before deleting them.

The WHERE CURRENT OF clause specifies a cursor pointing to a row that the DELETE command will remove. Once you define a cursor with DECLARE CURSOR and open a route with the OPEN command, you can use the cursor in a WHERE CURRENT OF clause to delete only the current row. Use the [FETCH](#) command to move the cursor to the next available row.

You must restore deleted rows from a backed up database or table. To recover disk space after deleting rows, use the [PACK](#) or [RELOAD](#) commands.

DELETE removes rows from a table or single-table view. If you have set transaction processing on, you can restore rows with [ROLLBACK](#). If not, you must restore them from a backup database or table. If you prefer not to use transaction processing, you can first use a relational command, such as [PROJECT](#), to make a backup copy of the table from which you are deleting rows. Then you can delete rows from the original table and remove the backup copy later.

Examples

The following command deletes all rows from the *transmastertable*. When you omit a [WHERE](#) clause, be sure that you want to delete all rows from the table.

```
DELETE FROM transmaster
```

The following command deletes rows from the *transmaster* table where the *custid* value is 100.

```
DELETE FROM transmaster WHERE custid = 100
```

1.6.6 DELETE DUPLICATES

Use the DELETE DUPLICATES command to remove duplicate rows from a table.

```
DELETE DUPLICATES FROM tblname [ USING collist ] ...  
... [ WHERE clause ]
```

Options

FROM tblname

Specifies the table name.

USING collist

Deletes rows based on duplicate values in the specified list of columns.

WHERE clause

Limits the rows of data to be deleted. For more information, see [WHERE](#).

About the DELETE DUPLICATES Command

Use DELETE DUPLICATES to delete duplicate rows from a table. A duplicate row is a row where the values for each column are exactly the same as those in another row in the table. This command deletes all but the first row for each set of duplicate rows.

DELETE DUPLICATES processes faster when the table contains an indexed column and the USING *collist* option is used.

Rules for Column Deletion

You can specify which rows to delete in a list of columns. The following rules apply:

- The first row is retained in the table.
- Any row with duplicate values in a specified column list is deleted, regardless of the values in any of its other columns.

Case Sensitivity

DELETE DUPLICATES is case sensitive when [CASE](#) is set on. For example, if CASE is set on and one row included the name SMITH and another row included the name Smith, R:BASE would not delete either row. However, if CASE was set off, R:BASE would delete the second row. (The default setting for CASE is off.)

NULL Values

When NULL values exist in the table, the [FONULL](#) setting must be set to ON to ensure duplicates are removed.

You must restore deleted rows from a backed up database or table. To recover the data's disk space after rows are deleted, use [PACK](#) or [RELOAD](#).

If you have set transaction processing on, you can restore rows with [ROLLBACK](#). If not, you must restore them from a backup database or table. If you prefer not to use transaction processing, you can first use a relational command, such as [PROJECT](#) to make a backup copy of the table from which you are deleting rows. Then you can delete rows from the original table and remove the backup copy later.

Example

The following command deletes duplicate rows from the *transmaster* table, but retains the first of the duplicate rows.

```
DELETE DUPLICATES FROM transmaster
```

The following example deletes duplicate rows based on the *transid*, *empid*, and *custid* columns in the *transmaster* table. Only the designated columns will be used to determine whether the rows are duplicates.

```
DELETE DUPLICATES FROM transmaster USING transid, empid, custid
```

The following deletes the duplicate rows based on the *transid* and *empid* columns in the *transmaster* table, where transaction dates are greater than January 1, 2020.

```
DELETE DUPLICATES FROM TransMaster +  
USING TransID, EmpID +  
WHERE TransDate > 01/01/2020)
```

1.6.7 DELETEOBJECT

Use the DELETEOBJECT command to delete form objects on demand.

```
DELETEOBJECT { <ComponentID> }  
              { <ParentControl> }
```

Options

ComponentID

The unique identifier that is assigned to the component, or control .

ParentControl

The form object in which other controls are contained within. A parent control can be an actual form or any of the following form controls: Panel, EnhancedPanel, GroupBox, EnhancedGroupBox, Tab Control, Enhanced Tab Control, Status Bar, RadioGroup, DB RadioGroup, Variable RadioGroup, Scrolling Region, etc.

This command can be directly associated to the [CREATEOBJECT](#) command.

1.6.8 DELROW

Use the DELROW command to delete the current row of data currently being edited or entered in a form.

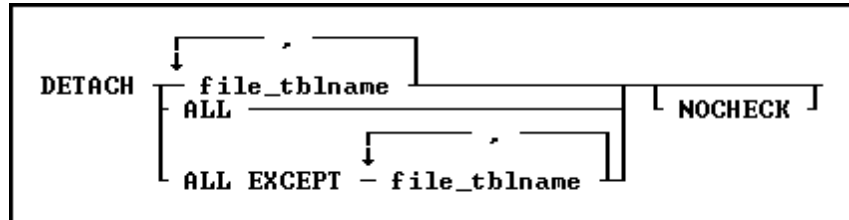
```
DELROW
```

About the DELROW Command

Use the DELROW command to delete the current row of data currently being edited or entered in a form.

1.6.9 DETACH

Use the DETACH command to remove a dBASE file table and its associated dBASE index files from the open R:BASE database.



Options

' Indicates that this part of the command is repeatable.

ALL

Removes all dBASE tables and associated dBASE index files from the open R:BASE database.

ALL EXCEPT file_tblname

Removes all dBASE tables and associated dBASE index files from the open R:BASE database, except the specified table.

file_tblname

Removes the specified dBASE table and associated dBASE index files from the open R:BASE database.

NOCHECK

Eliminates the confirmation message.

About the DETACH Command

Before you can detach a dBASE file table, an R:BASE database must be open. When you detach a dBASE file table, do not include the .DBF extension. DETACH requires the database owner's user identifier if one has been defined.

If you remove a dBASE file that is used in a form, report, label, or application, make the necessary changes to reflect detaching the dBASE file from the database. The rules, descriptions, and any access rights are deleted when you detach the dBASE file.

Use [ATTACH](#) to reattach a detached file to the same or a different database.

When [STATICDB](#) is set on, which activates a read-only schema mode, DETACH is only available for dBASE tables that were attached in the read-only schema mode, which creates temporary dBASE tables.

Example

In the following example, the first command line opens the *concomp* database. The second command line detaches the dBASE file table *sampgate* from the *concomp* database.

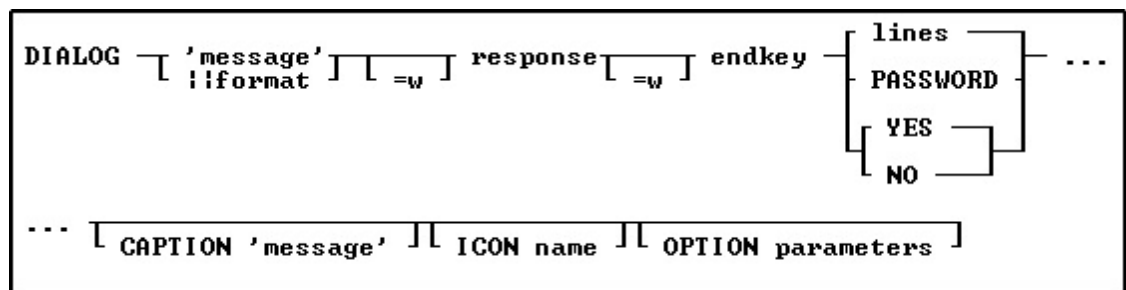
```

CONNECT concomp
DETACH sampgate

```

1.6.10 DIALOG

Use the DIALOG command to display a dialog box on the screen to accept text entry from a user. For assistance with building your DIALOG commands, refer to the [R:Dialog Builder Plugin](#), or the [R:Dialog with Buttons Builder Plugin](#).



Options

'message'

Specifies the text of the message to display in the dialog box. The value can also be passed as a variable. The limit is 2000 characters.

||format

Also known as EditMask. You can apply the mask specified by the EditMask parameter to the text string specified by the Value parameter. See the "[||format](#)" subchapter for more.

=w

Specifies the wrap width for the dialog box message.

response

Specifies the variable that will contain the dialog box entry. The input field allows for 100 characters per line.

=w

Specifies the width for the dialog box entry.

endkey

Specifies the variable containing the final keystroke ([Enter] or [Esc]) in a dialog box.

lines

Specifies the number of lines to display for the text entry in a dialog box. Up to 25 lines are allowed.

password

Displays the dialog box entry as asterisks. 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.

YES

Creates a Yes/No dialog box and sets the default response to Yes.

NO






Creates a Yes/No dialog box and sets the default response to No.

CAPTION 'message'

Specifies the text of the message to display in the dialog box caption. The value can also be passed as a variable.

ICON value

Icon "value" Parameter	Icon
APPS	
ATTENTION	
CONFIRM	

ERROR	
HELP	
INFO	
QUESTION	
SERIOUS	
STOP	
WARNING	
WINDOWS	 *

* will vary based upon Windows operating system

Additional **OPTION** parameters

Additional parameters are available to increase the visual display of the DIALOG window. To use the graphic DIALOG Builder, choose "Utilities" > "Plugins" > "DIALOG Builder" from the main Menu Bar. All OPTION parameters and values must be separated by the "|" (pipe) character.

OPTION Parameters	Description
Message	Changes how the DIALOG "message" is displayed
Input Field	Changes how the DIALOG "input field" is displayed
Buttons	Changes how the DIALOG "buttons" are displayed
Window	Changes how the DIALOG "window" is displayed
Popup Menu	Add a "custom popup menu" to the DIALOG window

Examples

Example 01: (Dialog with custom button text)

```
CLS
DIALOG 'Enter Last Name' vLastName=26 vEndKey 1 +
CAPTION 'Search Employee by Last Name' +
ICON 'APP' +
OPTION MESSAGE_FONT_COLOR BLACK +
|TRANSPARENCY 255 +
|WINDOW_BACK_COLOR WHITE +
|BUTTON_OK_CAPTION '&Search' +
|BUTTON_CANCEL_CAPTION '&Cancel'
```

Example 02: (Dialog with PASSWORD option)

```
CLS
DIALOG 'Enter Password' vPassword=26 vEndKey PASSWORD +
CAPTION 'Database Maintenance' +
ICON 'APP' +
OPTION MESSAGE_FONT_COLOR BLACK +
|TRANSPARENCY 255 +
|WINDOW_BACK_COLOR WHITE +
|BUTTON_OK_CAPTION '&Process' +
|BUTTON_CANCEL_CAPTION '&Cancel'
```

Example 03: (Dialog with Multi-Line Message with the [TOP LEFT](#) parameters)

```
-- (CHAR(009)) = Tab Key (Indent)
-- (CHAR(013)) = Carriage Return
CLS
SET VAR vMsg = +
('Line 1:'+(CHAR(009))+(CHAR(009))&'Contents of Line 1'+(CHAR(013)))+ +
'Line 2:'+(CHAR(009))+(CHAR(009))&'Contents of Line 2'+(CHAR(013))+ +
'Line 3:'+(CHAR(009))+(CHAR(009))&'Contents of Line 3'+(CHAR(013))+ +
'Line 4:'+(CHAR(009))+(CHAR(009))&'Contents of Line 4'+(CHAR(013))+ +
'Line 5:'+(CHAR(009))+(CHAR(009))&'Contents of Line 5'+(CHAR(013))+ +
'Line 6:'+(CHAR(009))+(CHAR(009))&'Contents of Line 6'+(CHAR(013))+ +
'Line 7:'+(CHAR(009))+(CHAR(009))&'Contents of Line 7'+(CHAR(013))+ +
'Line 8:'+(CHAR(009))+(CHAR(009))&'Contents of Line 8'+(CHAR(013))+ +
'Line 9:'+(CHAR(009))+(CHAR(009))&'Contents of Line 9'+(CHAR(013)))
DIALOG .vMsg vYesNo vEndKey YES +
CAPTION ' Your Dialog Caption Here ...' +
ICON 'APP' +
OPTION MESSAGE_FONT_COLOR BLACK +
|TRANSPARENCY 255 +
|WINDOW_BACK_COLOR WHITE +
|BUTTON_YES_CAPTION '&Start' +
|BUTTON_NO_CAPTION '&Cancel' +
|BUTTON_YES_COLOR GREEN +
|BUTTON_NO_COLOR RED +
|BUTTON_YES_FONT_COLOR WHITE +
|BUTTON_NO_FONT_COLOR WHITE +
|TOP 50 +
|LEFT 50
```

Example 04: (Dialog with button images and custom [message](#) format)

```
CLS
SET VAR vResponse TEXT = NULL
SET VAR vEndKey TEXT = NULL
DIALOG 'DIALOG Message Here ...' vResponse=26 vEndKey 1 +
CAPTION 'DIALOG Caption Here ...' ICON 'APPS' +
OPTION MESSAGE_FONT_COLOR GREEN +
|MESSAGE_FONT_NAME ARIAL +
|MESSAGE_BOLD OFF +
|WINDOW_BACK_COLOR WHITE +
|BUTTON_OK_CAPTION '&Continue' +
|BUTTON_CANCEL_CAPTION 'C&ancel' +
|BUTTON_YES_COLOR WHITE +
|BUTTON_NO_COLOR WHITE +
|BUTTON_YES_FONT_COLOR GREEN +
|BUTTON_NO_FONT_COLOR RED +
|BUTTONS_SHOW_GLYPH ON
```

Example 05: (Dialog with a custom [input field](#) background color with autoselect enabled)

```
DIALOG 'Please enter last name to search:' vResponse vEndKey 1 +
CAPTION 'DIALOG Window' ICON 'INFO' +
OPTION MESSAGE_FONT_COLOR 2147483647 +
|MESSAGE_FONT_NAME Tahoma +
|INPUT_FONT_COLOR RED +
|INPUT_FONT_SIZE 12 +
|INPUT_BACKGROUND_COLOR 9234160 +
```

```
|MESSAGE_BOLD ON +
|AUTOSELECT TRUE
```

Example 06: (Dialog window using Themes)

```
DIALOG 'DIALOG Message Here ...' vResponse=26 vEndKey 1 +
CAPTION 'DIALOG Caption Here ...' ICON 'APPS' +
OPTION THEMENAME Longhorn
```

Example 07: (Dialog with [Custom Buttons](#))

```
SET VAR vCaption TEXT = 'Run Monthly Routines'
SET VAR vYesNo TEXT = NULL
SET VAR vEndKey TEXT = NULL
CLS
DIALOG 'You have selected to run monthly routines' vYesNo vEndKey No +
CAPTION .vCaption +
ICON QUESTION +
OPTION WINDOW_BACK_COLOR WHITE +
|MESSAGE_BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_YES_CAPTION &Yes +
|BUTTON_YES_COLOR WHITE +
|BUTTON_YES_FONT_COLOR GREEN +
|BUTTON_YES_FONT_NAME Tahoma +
|BUTTON_YES_FONT_SIZE 11 +
|BUTTON_YES_FONT_BOLD ON +
|BUTTON_YES_WIDTH 85 +
|BUTTON_NO_CAPTION &No +
|BUTTON_NO_COLOR WHITE +
|BUTTON_NO_FONT_COLOR RED +
|BUTTON_NO_FONT_NAME Tahoma +
|BUTTON_NO_FONT_SIZE 11 +
|BUTTON_NO_FONT_BOLD ON +
|BUTTON_NO_WIDTH 85 +
|BUTTON_HEIGHT 35
IF vYesNo = 'No' OR vEndKey = '[Esc]' THEN
  GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
CLS
CLEAR VARIABLES vCaption,vYesNo,vEndKey
RETURN
```

Example 08: (Dialog with [cursor position](#) and R:BASE On-Screen Keyboard - ROSK):

```
SET VAR vResponse TEXT = 'Thank you'
DIALOG 'Message ...' vResponse vEndKey 1 +
CAPTION 'DIALOG Window' ICON INFO +
OPTION MESSAGE_FONT_COLOR 2147483647 +
|MESSAGE_BACK_COLOR -16777201+
|MESSAGE_FONT_NAME Tahoma+
|MESSAGE_FONT_SIZE 10+
|MESSAGE_BOLD ON+
|TOP 200 +
```



```
|LEFT 200 +
|CURSOR_POS 6 +
|ROSK_BUTTON ON
```

Example 09: (Dialog with Table Lookup [Popup](#))

```
DIALOG 'Enter Contact to Edit' vContactNo=26 vEndKey 1 +
CAPTION 'Contact Info' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for a List of Contacts' +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE LOOKUP +
|POPUP_TITLE_FONT_NAME VERDANA +
|POPUP_TITLE_FONT_SIZE 10 +
|POPUP_TITLE_FONT_BOLD TRUE +
|POPUP_TITLE_FONT_COLOR GREEN +
|POPUP_TITLE_BACK_COLOR WHITE +
|POPUP_TITLE_TEXT 'Select Contact to Edit' +
|POPUP_LIST_FONT_COLOR GREEN +
|POPUP_LIST_BACK_COLOR WHITE +
|POPUP_CAPTION 'List of Contacts' +
|POPUP_TABLE Contact +
|POPUP_SELECT ContFName,ContLName,ContPhone,ContEMail +
|POPUP_RETURN_COLUMN ContID +
|POPUP_WHERE ORDER BY ContFName +
|POPUP_SHOW_LINES TRUE +
|POPUP_RETURN_TYPE COLUMN +
|POPUP_LINES 20 +
|POPUP_DISTINCT TRUE
```

1.6.10.1 ||format

To use the Edit Mask feature with the DIALOG command, you must start the text string with || (double pipe character). The edit mask string consists of three fields with semicolons separating them. The **first part** of the mask is the mask itself. The **second part** is the character that determines whether the literal characters of the mask (dashes in social security number or parenthesis and hyphen for a phone number) are matched to characters of the first part or are inserted into the resulting value string. The **third part** of the mask is the character used to represent missing characters in the mask.

These are the special characters used in the **first part** of the mask:

Character	Meaning in mask
!	If a ! character appears in the mask, optional characters are represented in the returned string as leading blanks. If a ! character is not present, optional characters are represented in the returned string as trailing blanks.
>	If a > character appears in the mask, all characters that follow are in uppercase until the end of the mask or until a < character is encountered.
<	If a < character appears in the mask, all characters that follow are in lowercase until the end of the mask or until a > character is encountered.
<>	If these two characters appear together in a mask, no case checking is done.
\	The character that follows a \ character is a literal character. Use this character to use any of the mask special characters as a literal.
L	The L character requires an alphabetic character only in this position. For the US, this is A_Z, a_z.
	The character permits only an alphabetic character in this position, but doesn't require it.

A	The A character requires an alphanumeric character only in this position. For the US, this is A_Z, a_z, 0_9.
a	The a character permits an alphanumeric character in this position, but doesn't require it.
C	The C character requires an arbitrary character in this position.
c	The c character permits an arbitrary character in this position, but doesn't require it.
0	The 0 character requires a numeric character only in this position.
9	The 9 character permits a numeric character in this position, but doesn't require it.
#	The # character permits a numeric character or a plus or minus sign in this position, but doesn't require it.
:	The : character is used to separate hours, minutes, and seconds in times. If the character that separates hours, minutes, and seconds is different in the regional settings of the Control Panel, that character is substituted in the returned string.
/	The / character is used to separate months, days, and years in dates. If the character that separates months, days, and years is different in the regional settings of the Control Panel, that character is substituted in the returned string.
;	The ; character is used to separate the three fields of the mask.
_	The _ character automatically inserts spaces into the returned string.

Any character that does not appear in the preceding table can appear in the first part of the mask as a literal character. Literal characters are inserted automatically if the second field of the mask is 0, or matched to characters in the first part, if the second field is any other value. The special mask characters can also appear as literal characters if preceded by a backslash character (\).

The **second part** of the mask is a single character that indicates whether literal characters from the mask are included in the resulting value string. For example, the mask for a telephone number with area code could be the following string:

(000)_000_0000;0;*

The 0 in the second field indicates that the first part should consist of the 10 digits of the phone number, rather than the 14 characters that make up the final formatted string.

The 0 in the second field indicates that literals (dashes in social security number or parenthesis and hyphen for a phone number) are inserted into the resulting value string. Any other character besides zero indicates that that literals characters are included in the resulting value string.

The **third part** of the mask is the character that appears in the returned string for blanks. By default, this is the same as the character that stands for literal spaces. The two characters appear the same in the DIALOG. However, when a user edits the text in a masked edit control, the cursor selects each blank character in turn, and skips over the space character.

Note:

When working with multibyte character sets, such as Japanese Shift_JIS, each special mask character represents a single byte. To specify double_byte characters using the L,l,A,a,C, or c specifiers, the mask characters must be doubled as well. For example, LL would represent two single_byte alphabetic characters or a one double_byte character. Only single_byte literal characters are supported.

Typical Field Formats by Country:

United States	
Phone	4155551212 !\ (999\)000_0000;1;_
Extension	15450 !99999;1;_
Social Security No	555555555 000\ 00\ 0000;1;_
Short Zip Code	90504 00000;1;_
Long Zip Code	905040000 00000\ 9999;1;_
Date	062794 !99/99/00;1;_
Long Time	090515PM !90:00:00>LL;1;_

Short Time	1345 !90:00;1;_
------------	-------------------

United Kingdom	
Phone	1714561234 !\00009 0000099;1;_
Extension	15450 !99999;1;_
Postcode	WT3114TA >AAAa aaaa;1;_
Short Postcode	EC12 >AAAa;1;_
National Insurance	GM134152A !>AA000000A;1;_
Date	270195 !90/90/00;1;_
Long Time	090515PM !90:00:00>LL;1;_
Short Time	1345 !90:00;1;_

Taiwan	
Zip Code	107 000;1;_
Organization No	86692741 00000000;1;_
Social Security No	A324534422 L000000000;1;_
ROC Date	00850930 !9990.00.00;1;_
Date	100196 00/00/00;1;_
Phone	03532455768 \ (999\)9000 0000;1;_
Mobile Phone	090789879 000 000000;1;_
Pager	060345678 000 000000;1;_

Sweden	
Social Security No	6011012110 000000\ 0000;1;_
Organization Number	6011012110 000000\ 0000;1;_
Zip Code	64700 000 00;1;_
Date	950223 !00/00/00;1;_
Long Date	19950223 !0000/00/00;1;_
Long Time	210515 !90:00:00;1;_
Short Time	1345 !90:00;1;_

Spain	
Phone	4155551212 !\ (999\)0000000;1;_
Social Security No	123456789123456 0000000000000;1;_
Zip Code	90504 00000;1;_
Date	250895 !99/99/00;1;_
Long Time	210515 !90:00:00;1;_
Short Time	1345 !90:00;1;_

Norway	
Phone	22831310 00 09 99 99;1;_
Social Security No	01016512345 0000000 0000;1;_
Organization Number	01016512345 0000000 0000;1;_
Zip Code	0250 0000;1;_
Date	170595 00\00\ 00;1;_
Date	170595 00/00/00;1;_
Long Time	210515 !90:00:00;1;_
Short Time	1345 !90:00;1;_
Banking	12341212345 0000 00 00000;1;_

Netherland	
Phone	205405400 !\00099\ 0000999;1;_
Social Security No	123456789 000000000;1;_
Zip Code	1234AB 0000 >LL;1;_
Date	270695 !90/90/00;1;_
Long Time	214515 !90:00:00;1;_
Short Time	1345 !90:00;1;_

Korea	
Phone	025381890 !\ (9999\)000 0000;1;_
Handphone(Beeper)	0005381895 !000 000 0000;1;_

Social Security No	9507271234567 000000 0000000;1;
Zip Code	135080 000 000;1;
Date 1	950727 00 !90 90;1;
Date 2	19950727 !0000 !90 90;1;
Date 3	950727 00 ?!90-•90L ;1;
Date 4	19950727 !0000 ?!90-•90L ;1;
Long Time	042520PM !90:00:00 >LL;1;
Long Korean time	042520PM !90 +00 00+ >LL;1;
Short Time	1625 !90:00;1;
Short Korean Time	1625 !90 +00 ;1;

Japan	
Phone	0353508393 !099999 0999 0000;1;
Zip Code	1510073 000\ 0000;1;
Date	062794 !99/99/00;1;
Time	090515PM !90:00:00>LL;1;

Italy	
Phone	0212345678 !\ (9999\) 0000 0000;1;
Phone with Extension	02123456781234 !\ (9999\) 0000 0000\ .9999;1;
Local Phone	12345678 !0000 0000;1;
Company Tax #	00010010023 000000000000;1;
Tax Payer #	TRMDNC55P27B242Z >LLLLLL00L00L000L;1;
Zip Code	00100 00000;1;
Short Date	130295 !99/99/00;1;
Long Date 1	130295 !99/99/\1\900;1;
Long Date 2	13Gen95 !99 >L<LL \1\900;1;
Long Time	210515 !90:00:00;1;
Short Time	1345 !90:00;1;

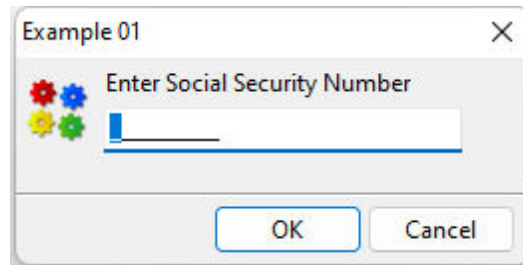
Germany	
Phone	06103979 !\ (999 99\) 99999;1;
Extension	2010 ! 9999;1;
Banking	50850150 000\ 000\ 00;1;
BRD ZIP Code	63225 D 00000;1;
Intl ZIP Code	CH50000 >cc 99999;1;
Date	301193 !90/90/00;1;
Short Time	1345 !90:00;1;
Long Time	193059 !90:00:00;1;

France	
Telephone	0299887766 !00\ 00\ 00\ 00\ 00;1;
Bank Card	5555444433332222 !0000 0000 0000 0000;1;
Social Security No	166107811122233 !00000000000000\ 99;1;
Zip Code	92130 !00000;1;
Short Date	2712 !99/99;1;
Long Date 1	271296 !99/99/00;1;
Long Date 2	27Déc1996 !99 >L<LL 0000;1;
Long Time	134515 !90:00:00;1;
Short Time	1345 !90:00;1;

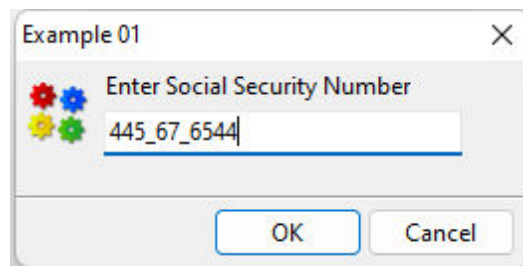
Denmark	
Phone	48140001 00 09 99 99;1;
Phone with Country	48140001 \+45 00 00 00 00;1;
Social Security No	1234567890 000000\ 0000;1;
Zip Code	3450 0000;1;
Zip Code with Country	DK3450 !>LL\ 0000;1;
Date	260195 90\ .90\ .\1\900;1;
Date	260195 90/90/00;1;
Long Time	210515 !90:00:00;1;
Short Time	1345 !90:00;1;

Example 01: (Capturing Social Security Number)

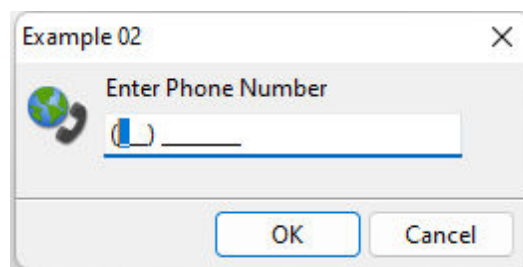
```
SET VAR vSSN TEXT = NULL
CLS
DIALOG 'Enter Social Security Number||000_00_0000;0;_' +
vSSN=20 vEndKey 1 CAPTION 'Example 01' ICON 'APPS'
```

**Example 01a:** (Capturing Social Security Number)

```
SET VAR vSSN TEXT = NULL
SET VAR vMessage TEXT = 'Enter Social Security Number||000_00_0000;0;_'
SET VAR vCaption TEXT = 'Example 01a'
CLS
DIALOG .vMessage vSSN=20 vEndKey 1 CAPTION .vCaption ICON 'APPS'
```

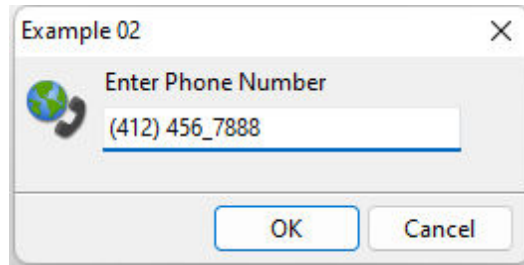
**Example 02:** (Capturing Phone Number)

```
SET VAR vPhone TEXT = NULL
CLS
DIALOG 'Enter Phone Number||(000) 000_0000;0;_' +
vPhone=20 vEndKey 1 CAPTION 'Example 02' ICON 'HELP'
```

**Example 02a:** (Capturing Phone Number)

```
SET VAR vPhone TEXT = NULL
SET VAR vMessage TEXT = 'Enter Phone Number||(000) 000_0000;0;_'
```

```
SET VAR vCaption TEXT = 'Example 02a'
CLS
DIALOG .vMessage vPhone=20 vEndKey 1 CAPTION .vCaption ICON 'HELP'
```



Example 03: (Capturing Zip Code)

```
SET VAR vZipCode TEXT = NULL
CLS
DIALOG 'Enter Zip Code||00000_0000;0;_' +
vZipCode=20 vEndKey 1 CAPTION 'Example 03' ICON 'QUESTION'
```

Example 03a: (Capturing Zip Code)

```
SET VAR vZipCode TEXT = NULL
SET VAR vMessage TEXT = 'Enter Zip Code||00000_0000;0;_'
SET VAR vCaption TEXT = 'Example 03a'
CLS
DIALOG .vMessage vZipCode=20 vEndKey 1 CAPTION .vCaption ICON 'QUESTION'
```

Example 04: (Capturing Credit Card Number)

```
SET VAR vCCNumber TEXT = NULL
CLS
DIALOG 'Enter Credit Card Number||0000_0000_0000_0000;0;_' +
vCCNumber=20 vEndKey 1 CAPTION 'Example 04' ICON 'WINDOWS'
```

Example 04a: (Capturing Credit Card Number)

```
SET VAR vCCNumber TEXT = NULL
SET VAR vMessage TEXT = 'Enter Credit Card Number||0000_0000_0000_0000;0;_'
SET VAR vCaption TEXT = 'Example 04a'
CLS
DIALOG .vMessage vCCNumber=20 vEndKey 1 CAPTION .vCaption ICON 'WINDOWS'
```

1.6.10.2 OPTION parameters

These parameters can be used to change the "look and feel" of DIALOG window. All parameters and values must be separated by the "|" (pipe) symbol.

OPTION Parameters	Description
Message	Changes how the DIALOG "message" is displayed
Input Field	Changes how the DIALOG "input field" is displayed
Buttons	Changes how the DIALOG "buttons" are displayed
Window	Changes how the DIALOG "window" is displayed
Popup Menu	Add a "custom popup menu" to the the DIALOG window

1.6.10.2.1 Message

These parameters are specific to changing how the DIALOG "message" is displayed.

Parameter	Value	Description
MESSAGE_FONT_COLOR	value	Changes the font color of message area. User can specify the integer value or type one of the predefined color names.
MESSAGE_BACK_COLOR	value	Changes the background color of message area. User can specify the integer value or type one of the predefined color names.
MESSAGE_BOLD	ON OFF	Makes font in the message area bold style.
MESSAGE_ITALIC	ON OFF	Makes font in the message area italic style.
MESSAGE_UNDERLINE	ON OFF	Makes font in the message area underlined.
MESSAGE_STRIKEOUT	ON OFF	Makes font in the message area strikeout.
MESSAGE_FONT_NAME	value	Specifies a font name in the message area.
MESSAGE_FONT_SIZE	value	Specifies a font size (integer value) in the message area.
MESSAGE_SHOW_ACCELERATOR_CHAR	ON OFF	Specifies if the message displays the accelerator character. The default value is ON.

Examples:

```
DIALOG 'Message ...' vResponse vEndKey 1 +
CAPTION 'DIALOG Window' ICON 'INFO' +
OPTION MESSAGE_FONT_COLOR BLUE +
|MESSAGE_BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_BOLD ON
```

```
DIALOG +
'You have selected to process new sales order for Fenwick & Sons Glass' +
vYesNo vEndKey Yes +
CAPTION 'New Sales Order' ICON QUESTION +
OPTION WINDOW_BACK_COLOR WHITE +
|MESSAGE_BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 11 +
|BUTTON_YES_CAPTION &Yes +
|BUTTON_YES_COLOR WHITE +
|BUTTON_YES_FONT_COLOR GREEN +
|BUTTON_NO_CAPTION &No +
|BUTTON_NO_COLOR WHITE +
|BUTTON_NO_FONT_COLOR RED +
|MESSAGE_SHOW_ACCELERATOR_CHAR OFF
```

1.6.10.2.2 Input Field

These parameters are specific to changing how the DIALOG "input field" is displayed.

Parameter	Value	Description
INPUT_FONT_NAME	value	Specifies a font name in the input field.

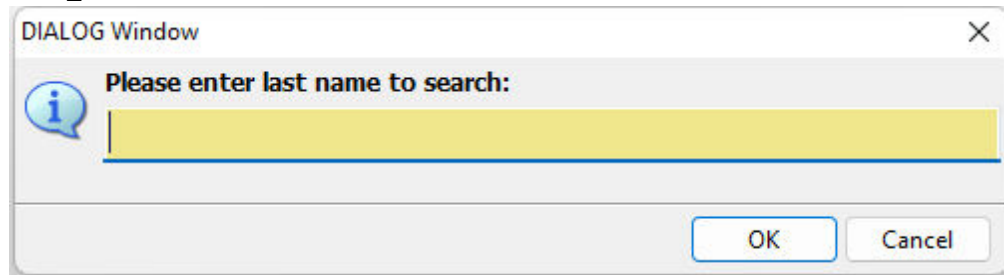
INPUT_FONT_COLOR	value	Changes the font color of input field. User can specify the integer value or type one of the predefined color names.
INPUT_FONT_SIZE	OFF SMALL	Specifies a font size (integer value) in the input field
INPUT_BACKGROUND_COLOR	value	Changes the background color of input field. User can specify the integer value or type one of the predefined color names.
DIALOG_EDIT_HINT	TRUE FALSE	Specifies the hint when hovering the mouse cursor over the dialog edit box. The option is useful to provide a hint, when a popup menu is defined.
AUTOSELECT	TRUE FALSE	Specifies the entire value within the field is selected

Examples:

```

DIALOG 'Please enter last name to search:' vResponse vEndKey 1 +
CAPTION 'DIALOG Window' ICON 'INFO' +
OPTION MESSAGE_FONT_COLOR 2147483647 +
|MESSAGE_FONT_NAME Tahoma +
|INPUT_FONT_COLOR RED +
|INPUT_FONT_SIZE 12 +
|INPUT_BACKGROUND_COLOR 9234160 +
|MESSAGE_BOLD ON

```



```

-- Using_Input_Color_For_Dialog.RMD
-- Author: A. Razzak Memon
SET VAR vInput TEXT = NULL
SET VAR vEndKey TEXT = NULL
SET VAR vCaption TEXT = 'Using Input Color for DIALOG Box'
LABEL GetInput
CLS
SET VAR vInput = NULL
DIALOG 'Enter Message Text' vInput=30 vEndKey 1 +
CAPTION .vCaption ICON 'INFO' +
OPTION MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR NAVY +
|MESSAGE_FONT_SIZE 11 +
|INPUT_BACKGROUND_COLOR YELLOW +
|THEMENAME Longhorn
IF vEndKey = '[Esc]' THEN
GOTO Done
ENDIF
IF vInput IS NULL THEN
PAUSE 2 USING 'Missing Message Text' +
CAPTION .vCaption ICON 'WARNING' +
BUTTON 'Press any key to enter message text' +
OPTION MESSAGE_FONT_NAME Tahoma +

```



```

|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 11 +
|INPUT_BACKGROUND_COLOR YELLOW +
|THEMENAME Longhorn
GOTO GetInput
ENDIF
-- Do what you have to do here ...
LABEL Done
CLEAR VARIABLES vInput,vEndKey,vCaption
RETURN

```

1.6.10.2.3 Buttons

These parameters are specific to changing how the DIALOG "buttons" are displayed.

Since you can use a DIALOG window with an edit field using the [OK and CANCEL](#) buttons, as well as only [YES and NO](#) buttons without the edit field, there are several "Button" parameters available.

The follow parameters can be used to alter both buttons at the same time.

Parameter	Value	Description
BUTTONS_SHOW_GLYPH	ON OFF	Specifies if images are displayed on the "OK and Cancel" or "Yes and No" buttons.
BUTTONS_BACK_COLOR	value	Specifies the background color for the panel where the buttons are located.

Additional parameters are available within the subchapters of this topic to alter the button properties individually.

Example:

```

DIALOG 'DIALOG Message Here ...' vResponse=26 vEndKey 1 +
CAPTION 'DIALOG Caption Here ...' ICON 'APPS' +
OPTION WINDOW_CAPTION ON +
|WINDOW_BACK_COLOR WHITE +
|BUTTONS_BACK_COLOR WHITE +
|BUTTON_OK_CAPTION '&Continue' +
|BUTTON_CANCEL_CAPTION 'C&ancel' +
|BUTTON_OK_FONT_COLOR GREEN +
|BUTTON_CANCEL_FONT_COLOR RED

```

1.6.10.2.3.1 OK and CANCEL

These parameters are specific to changing how the DIALOG **OK** and **CANCEL** buttons are displayed.

Parameter	Value	Description
BUTTON_OK_CAPTION	value	Specifies the text for the OK button.
BUTTON_OK_FONT_COLOR	value	Specifies the font color for OK button.
BUTTON_OK_BACK_COLOR	value	Specifies the background color for OK button.
BUTTON_OK_COLOR	value	Specifies the color for OK button
BUTTON_OK_FONT_BOLD	ON OFF	Specifies if the OK button font is bold type.
BUTTON_OK_FONT_ITALIC	ON OFF	Specifies if the OK button font is italic type.
BUTTON_OK_FONT_UNDERLINE	ON OFF	Specifies if the OK button font is underline type.

BUTTON_OK_FONT_STRIKEOUT	ON OFF	Specifies if the OK button font is strike out type.
BUTTON_OK_FONT_NAME	value	Specifies the font name for OK button.
BUTTON_OK_FONT_SIZE	value	Specifies the font size for OK button.
BUTTON_OK_WIDTH	value	Specifies the OK button width in pixels.
BUTTON_CANCEL_CAPTION	value	Specifies the text for the Cancel button.
BUTTON_CANCEL_FONT_COLOR	value	Specifies the font color for the Cancel button.
BUTTON_CANCEL_BACK_COLOR	value	Specifies the background color for Cancel button.
BUTTON_CANCEL_COLOR	value	Specifies the color for Cancel button.
BUTTON_CANCEL_FONT_BOLD	ON OFF	Specifies if the Cancel button font is bold type.
BUTTON_CANCEL_FONT_ITALIC	ON OFF	Specifies if the Cancel button font is italic type.
BUTTON_CANCEL_FONT_UNDERLINE	ON OFF	Specifies if the Cancel button font is underline type.
BUTTON_CANCEL_FONT_STRIKEOUT	ON OFF	Specifies if the Cancel button font is strike out type.
BUTTON_CANCEL_FONT_NAME	ON OFF	Specifies the font name for Cancel button.
BUTTON_CANCEL_FONT_SIZE	value	Specifies the font size for Cancel button.
BUTTON_CANCEL_WIDTH	value	Specifies the Cancel button width in pixels.
BUTTON_HEIGHT	value	Specifies the height of the buttons in pixels.

1.6.10.2.3.2 YES and NO

These parameters are specific to changing how the DIALOG **YES** and **NO** buttons are displayed.

Parameter	Value	Description
BUTTON_YES_BACK_COLOR	value	Specifies the background color for the Yes button.
BUTTON_YES_CAPTION	value	Specifies the text for the Yes button.
BUTTON_YES_FONT_SIZE	value	Specifies the font size for Yes button.
BUTTON_YES_FONT_BOLD	ON OFF	Specifies if the Yes button font is bold type.
BUTTON_YES_FONT_ITALIC	ON OFF	Specifies if the Yes button font is italic type.
BUTTON_YES_FONT_NAME	value	Specifies the font name for Yes button.
BUTTON_YES_FONT_STRIKEOUT	ON OFF	Specifies if the Yes button font is strike out type.
BUTTON_YES_FONT_UNDERLINE	ON OFF	Specifies if the Yes button font is underline type.
BUTTON_YES_WIDTH	value	Specifies the Yes button width in pixels.
BUTTON_YES_FONT_COLOR	value	Specifies the font color for Yes button.
BUTTON_YES_COLOR	value	Specifies the color for Yes button.
BUTTON_NO_CAPTION	value	Specifies the text for the No button.
BUTTON_NO_FONT_SIZE	value	Specifies the font size for No button.
BUTTON_NO_FONT_BOLD	ON OFF	Specifies if the No button font is bold type.
BUTTON_NO_FONT_ITALIC	ON OFF	Specifies if the No button font is italic type.
BUTTON_NO_FONT_NAME	value	Specifies the font name for No button.
BUTTON_NO_FONT_STRIKEOUT	ON OFF	Specifies if the No button font is strike out type.
BUTTON_NO_FONT_UNDERLINE	ON OFF	Specifies if the No button font is underline type.
BUTTON_NO_WIDTH	value	Specifies the No button width in pixels.

BUTTON_NO_FONT_COLOR	value	Specifies the font color for No button.
BUTTON_NO_COLOR	value	Specifies the color for No button.
BUTTON_NO_BACK_COLOR	value	Specifies the background color for NO button.
BUTTON_HEIGHT	value	Specifies the height of the buttons in pixels.

```
-- Example (DIALOG with Custom Buttons)
SET VAR vCaption TEXT = 'Run Monthly Routines'
SET VAR vYesNo TEXT = NULL
SET VAR vEndKey TEXT = NULL
CLS
DIALOG 'You have selected to run monthly routines' vYesNo vEndKey No +
CAPTION .vCaption +
ICON QUESTION +
OPTION WINDOW_BACK_COLOR WHITE +
|MESSAGE_BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_YES_CAPTION &Yes +
|BUTTON_YES_COLOR WHITE +
|BUTTON_YES_FONT_COLOR GREEN +
|BUTTON_YES_FONT_NAME Tahoma +
|BUTTON_YES_FONT_SIZE 11 +
|BUTTON_YES_FONT_BOLD ON +
|BUTTON_YES_WIDTH 85 +
|BUTTON_NO_CAPTION &No +
|BUTTON_NO_COLOR WHITE +
|BUTTON_NO_FONT_COLOR RED +
|BUTTON_NO_FONT_NAME Tahoma +
|BUTTON_NO_FONT_SIZE 11 +
|BUTTON_NO_FONT_BOLD ON +
|BUTTON_NO_WIDTH 85
IF vYesNo = 'No' OR vEndKey = '[Esc]' THEN
GOTO Done
ENDIF
-- Do what you have to do here ...
LABEL Done
CLS
CLEAR VARIABLES vCaption,vYesNo,vEndKey
RETURN
```

1.6.10.2.4 Window

These parameters are specific to changing how the DIALOG "window" is displayed.

Parameter	Value	Description
MODERN_LOOK	ON OFF	Specifies to use the modern look of recent Windows Aero releases. The modern look performs automatic message wrapping for large content at 450 pixels. Please note: The custom formatting used in existing DIALOG commands will be ignored.
ICON_FILE	value	Specifies the external path and name for an image file.

		<p>NOTE: Any of the below image file formats can be used. There is no size limitation, though 32x32 is recommended. The image can be resized using the RESIZE_ICON parameter. To display custom BMP images with transparency, use the first pixel, which is on the bottom left corner, as the transparent color. Any pixel with that color becomes transparent.</p> <table><tr><td>wmf</td><td>Windows Metafiles</td></tr><tr><td>emf</td><td>Enhanced Metafiles</td></tr><tr><td>ico</td><td>Icon Files</td></tr><tr><td>bmp</td><td>Windows Bitmap Files</td></tr><tr><td>png</td><td>Portable Network Graphics</td></tr><tr><td>jpeg, jpg</td><td>JPEG Image File</td></tr><tr><td>tif, tiff</td><td>TIFF Image File</td></tr><tr><td>gif</td><td>Compuserve GIF Image</td></tr><tr><td>jp2, j2k, jpc</td><td>JPEG 2000 Image</td></tr><tr><td>tga, afi</td><td>Truevision Targa File</td></tr><tr><td>vst</td><td>Truevision Vista File</td></tr><tr><td>pcx, pcc</td><td>ZSoft PC Paintbrush</td></tr><tr><td>dcx</td><td>DCX Image File</td></tr></table>	wmf	Windows Metafiles	emf	Enhanced Metafiles	ico	Icon Files	bmp	Windows Bitmap Files	png	Portable Network Graphics	jpeg, jpg	JPEG Image File	tif, tiff	TIFF Image File	gif	Compuserve GIF Image	jp2, j2k, jpc	JPEG 2000 Image	tga, afi	Truevision Targa File	vst	Truevision Vista File	pcx, pcc	ZSoft PC Paintbrush	dcx	DCX Image File
wmf	Windows Metafiles																											
emf	Enhanced Metafiles																											
ico	Icon Files																											
bmp	Windows Bitmap Files																											
png	Portable Network Graphics																											
jpeg, jpg	JPEG Image File																											
tif, tiff	TIFF Image File																											
gif	Compuserve GIF Image																											
jp2, j2k, jpc	JPEG 2000 Image																											
tga, afi	Truevision Targa File																											
vst	Truevision Vista File																											
pcx, pcc	ZSoft PC Paintbrush																											
dcx	DCX Image File																											
RESIZE_ICON	ON OFF	Specifies to resize the image (except .ico) to 32x32																										
ROSK_BUTTON	ON OFF	Specifies to display a button to launch the R:BASE On-Screen Keyboard (ROSK)																										
CURSOR_POS	value	Specifies the cursor position, for the specified text in the edit field.																										
WINDOW_CAPTION	OFF SMALL	Customizes the window caption. The OFF value makes window caption invisible. The SMALL value makes window caption small (tool window).																										
WINDOW_BACK_COLOR	value	To change the background color of window area. User can specify the integer value or type one of the predefined color names.																										
TRANSPARENCY	0 - 255	Set the transparency to specify a layered window with a translucent color. The transparency value specifies the degree of translucency.																										
TOP	value	Moves the dialog box location, in pixels, from the top of the screen down.																										
LEFT	value	Moves the dialog box location, in pixels, from the left of the screen over.																										
BOTTOM	value	Moves the dialog box location, in pixels, from the bottom of the screen up.																										
RIGHT	value	Moves the dialog box location, in pixels, from the right of the screen over.																										
THEMENAME	value	<p>Specifies one of 86 pre-defined Themes, or a custom Theme loaded into R:BASE. See Themes.</p> <p>NOTE: All previously defined DIALOG commands which are redefined to use Themes should be thoroughly checked prior to putting into production as objects and text may look substantially different.</p>																										

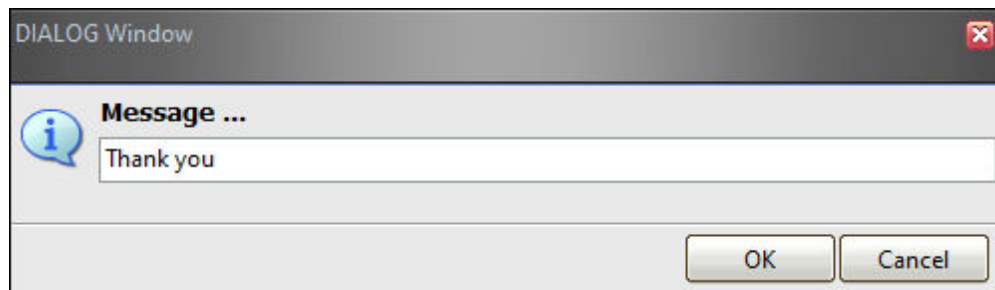
Example (Cursor Position):

```
SET VAR vResponse TEXT = 'Thank you'
```

```

DIALOG 'Message ...' vResponse vEndKey 1 +
CAPTION 'DIALOG Window' ICON INFO +
OPTION MESSAGE_FONT_COLOR 2147483647 +
|MESSAGE_BACK_COLOR -16777201+
|MESSAGE_FONT_NAME Tahoma+
|MESSAGE_FONT_SIZE 10+
|MESSAGE_BOLD ON+
|TOP 200 +
|LEFT 200 +
|THEMENAME Longhorn +
|CURSOR_POS 6

```



Example (Custom Image):

```

DIALOG 'Do you wish to continue?' vResponse vEndKey YES +
CAPTION 'Confirm' OPTION ICON_FILE Logo128x128.jpg|RESIZE_ICON ON +
|MESSAGE_FONT_COLOR NAVY|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_SIZE 10|MESSAGE_BOLD ON

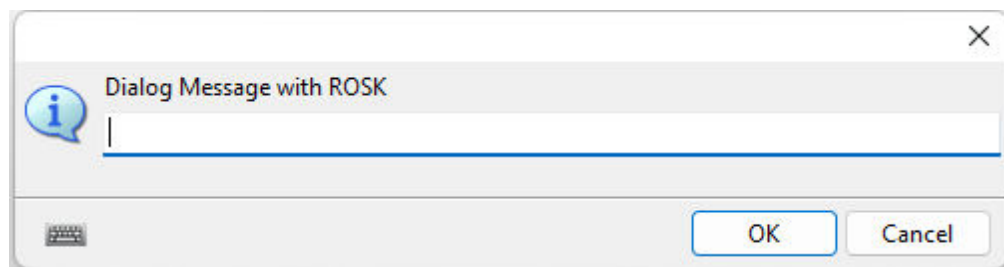
```

Example (R:BASE On-Screen Keyboard (ROSK)):

```

DIALOG 'Dialog Message with ROSK' vResponse vEndKey 1 +
CAPTION ' ' OPTION ROSK_BUTTON ON

```

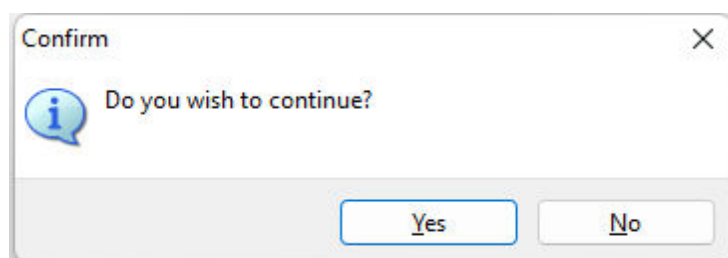


Example (Modern Look):

```

DIALOG 'Do you wish to continue?' vResponse vEndKey YES +
CAPTION 'Confirm' OPTION MODERN_LOOK ON

```



1.6.10.2.5 Popup Menu

These parameters are specific to adding a "custom popup menu" to the the DIALOG window.

Parameter	Value	Description
POPUP_ENABLED	ON OFF (default)	Enables the option of displaying a custom pop-up menu when the user double-clicks within the data field
POPUP_DIALOG_TYPE	LOOKUP DATE TIME DATETIME CALCULATOR	Specifies the Pop-up type when double-clicking on the DIALOG box.
DIALOG_EDIT_HINT	value	Specifies the hint when hovering the mouse cursor over the dialog edit box. The option is useful to provide a hint, when a popup menu is defined.
AUTOSELECT	TRUE FALSE	Specifies the entire value within the field is selected

1.6.10.2.5.1 LOOKUP

These parameters are specific to adding a "LOOKUP" custom popup menu to the the DIALOG window.

Parameter	Value	Description
POPUP_TABLE	value	Specifies the table/view name for the pop-up menu.
POPUP_SELECT	value	Specifies the SELECT clause for the pop-up menu values. e.g. <code>Company, CustState, CustZip</code>
POPUP_WHERE	value	Specifies the WHERE clause for the pop-up menu values. e.g. <code>WHERE CustState = 'PA' ORDER BY Company ASC</code>
POPUP_DISTINCT	TRUE FALSE	Specifies the option to display distinct values for selected columns.
POPUP_RETURN_COLUMN	value	Specifies the column returned. The POPUP_RETURN_COLUMN parameter must be located after your WHERE Clause parameter, if exists, otherwise R:BASE will provide all values specified in your POPUP_SELECT portion of the command.
POPUP_RETURN_TYPE	VALUE COLUMN	Specifies the option to return either the selected item in pop-up menu or the value of POPUP_RETURN_COLUMN.
POPUP_CAPTION	value	Specifies the text for custom pop-up window caption.
POPUP_TITLE_FONT_NAME	value	Specifies the font name for custom pop-up title.
POPUP_TITLE_FONT_SIZE	value	Specifies the font size for custom pop-up title.
POPUP_TITLE_FONT_COLOR	value	Specifies the font color for custom pop-up title.
POPUP_TITLE_FONT_BOLD	ON OFF	Specifies the bold font for custom pop-up title.
POPUP_TITLE_FONT_ITALIC	ON OFF	Specifies the italic font for custom pop-up title.
POPUP_TITLE_FONT_UNDERLINE	ON OFF	Specifies the underline font for custom pop-up title.
POPUP_TITLE_FONT_STRIKEOUT	ON OFF	Specifies the strike out font for custom pop-up title.

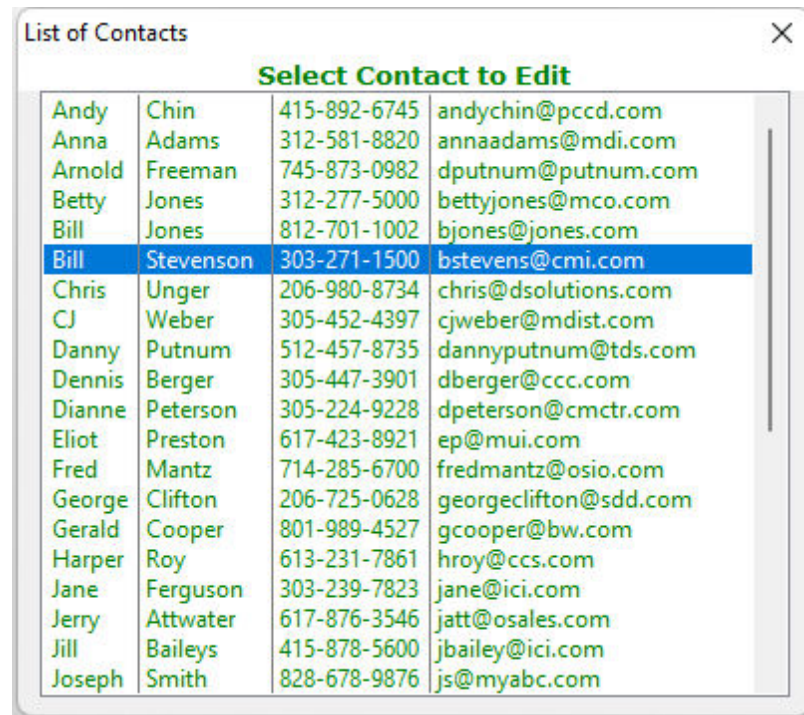
POPUP_TITLE_TEXT	value	Specifies the text for custom pop-up title.
POPUP_TITLE_BACK_COLOR	value	Specifies the background color for custom pop-up title.
POPUP_LIST_FONT_NAME	value	Specifies the font name for custom pop-up list items.
POPUP_LIST_FONT_SIZE	value	Specifies the font size for custom pop-up list items.
POPUP_LIST_FONT_COLOR	value	Specifies the font color for custom pop-up list items.
POPUP_LIST_FONT_BOLD	ON OFF	Specifies the bold font for custom pop-up list items.
POPUP_LIST_FONT_ITALIC	ON OFF	Specifies the italic font for custom pop-up list items.
POPUP_LIST_FONT_UNDERLINE	ON OFF	Specifies the underline font for custom pop-up list items.
POPUP_LIST_FONT_STRIKEOUT	ON OFF	Specifies the strike out font for custom pop-up list items.
POPUP_LIST_BACK_COLOR	value	Specifies the background color for custom pop-up list items.
POPUP_LIST_MULTISELECT	TRUE FALSE	Enables the option to select the multiple values within the pop-up list items.
POPUP_LINES	value	Specifies the number of visible lines to be displayed for the list items.
POPUP_SHOW_LINES	TRUE FALSE	Specifies the option to show lines between selected columns for the list items.

Example:

```

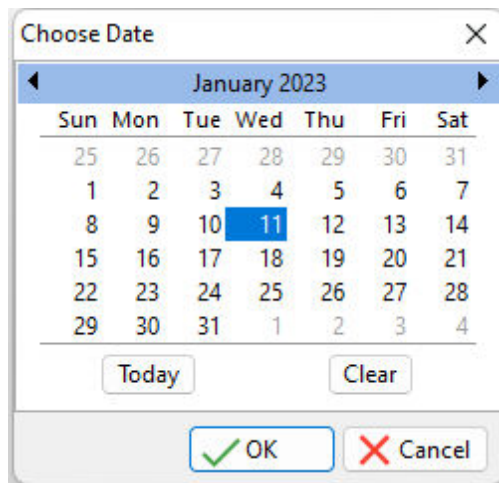
DIALOG 'Enter Contact to Edit' vContactNo=26 vEndKey 1 +
CAPTION 'Contact Info' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for a List of Contacts' +
| POPUP_ENABLED TRUE +
| POPUP_DIALOG_TYPE LOOKUP +
| POPUP_TITLE_FONT_NAME VERDANA +
| POPUP_TITLE_FONT_SIZE 10 +
| POPUP_TITLE_FONT_BOLD TRUE +
| POPUP_TITLE_FONT_COLOR GREEN +
| POPUP_TITLE_BACK_COLOR WHITE +
| POPUP_TITLE_TEXT 'Select Contact to Edit' +
| POPUP_LIST_FONT_COLOR GREEN +
| POPUP_LIST_BACK_COLOR WHITE +
| POPUP_CAPTION 'List of Contacts' +
| POPUP_TABLE Contact +
| POPUP_SELECT ContFName,ContLName,ContPhone,ContEMail +
| POPUP_RETURN_COLUMN ContID +
| POPUP_WHERE ORDER BY ContFName +
| POPUP_SHOW_LINES TRUE +
| POPUP_RETURN_TYPE COLUMN +
| POPUP_LINES 20 +
| POPUP_DISTINCT TRUE

```



1.6.10.2.5.2 DATE

These parameters are specific to adding a "DATE" custom popup menu to the the DIALOG window. The font settings for the calendar will be based upon the [input field](#) parameters: INPUT_FONT_NAME, INPUT_FONT_COLOR, INPUT_FONT_SIZE.



Parameter	Value	Description
POPUP_CALENDAR_BACK_COLOR	value	Specifies the background color of the calendar.
POPUP_CALENDAR_DAYS_COLOR	value	Specifies the color of the calendar days.
POPUP_CALENDAR_FILLDAYS_COLOR	value	Specifies the color of the calendar fill days.
POPUP_CALENDAR_DAYSOFWEEK_COLOR	value	Specifies the color of the calendar days of week.
POPUP_CALENDAR_LINES_COLOR	value	Specifies the color of the calendar lines.

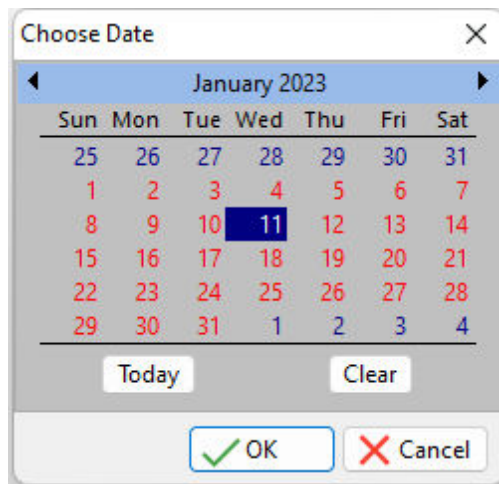
POPUP_CALENDAR_SELBACK_COLOR	value	Specifies the color of the selected date background.
POPUP_CALENDAR_SELFORE_COLOR	value	Specifies the color of the selected date foreground.
POPUP_CALENDAR_TODAYFRAME_COLOR	value	Specifies the color of today's frame.

Example:

```

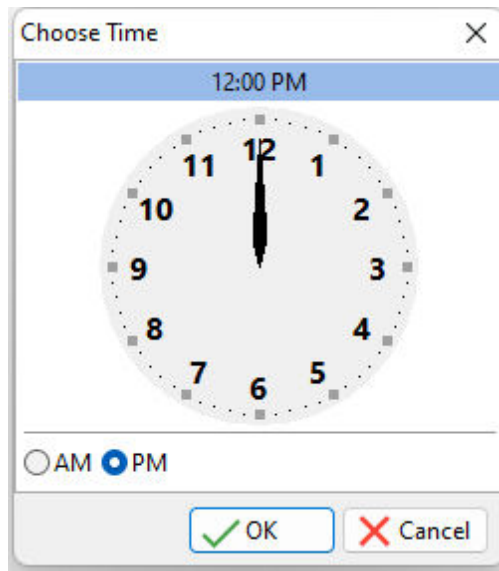
DIALOG 'Enter Date to Browse:' vBrowseDate=26 vEndKey 1 +
CAPTION 'Enter Date' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for Calendar' +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE DATE +
|POPUP_CALENDAR_BACK_COLOR SILVER +
|POPUP_CALENDAR_DAYS_COLOR RED +
|POPUP_CALENDAR_FILLDAYS_COLOR NAVY +
|POPUP_CALENDAR_DAYSOFWEEK_COLOR +
|POPUP_CALENDAR_LINES_COLOR BLACK +
|POPUP_CALENDAR_SELBACK_COLOR NAVY +
|POPUP_CALENDAR_SELFORE_COLOR WHITE +
|POPUP_CALENDAR_TODAYFRAME_COLOR RED

```



1.6.10.2.5.3 TIME

These parameters are specific to adding a "TIME" custom popup menu to the the DIALOG window. The font settings for the clock will be based upon the [input field](#) parameters: INPUT_FONT_NAME, INPUT_FONT_COLOR, INPUT_FONT_SIZE.



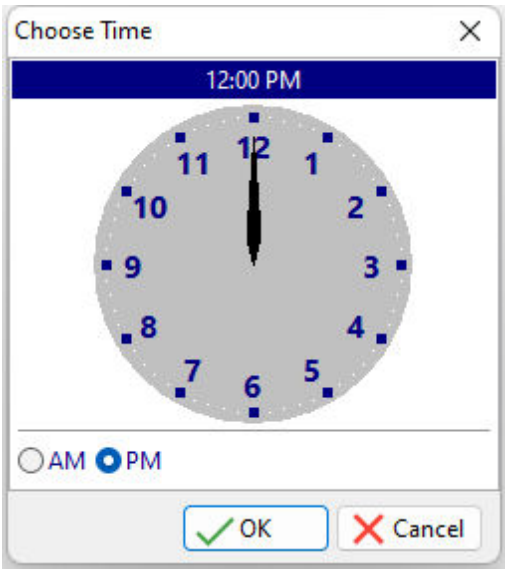
Parameter	Value	Description
POPUP_CLOCK_BACK_COLOR	value	Specifies the background color of the clock.
POPUP_CLOCK_FACE_COLOR	value	Specifies the face color of the clock.
POPUP_CLOCK_HANDS_COLOR	value	Specifies the color of the clock's hands.
POPUP_CLOCK_NUMBERS_COLOR	value	Specifies the color of the clock's numbers.
POPUP_CLOCK_HOURLTICKS_COLOR	value	Specifies the color of the clock's hour ticks.
POPUP_CLOCK_MINUTETICKS_COLOR	value	Specifies the color of the clock's minute ticks.
POPUP_CLOCK_HEADER_COLOR	value	Specifies the background color of the clock header.
POPUP_CLOCK_HEADER_FONT_COLOR	value	Specifies the font color of the clock header.
POPUP_CLOCK_AMPM_FONT_COLOR	value	Specifies the font color of the AM/PM radio buttons.
POPUP_CLOCK_DEFAULT_TIME	value	Specifies the default time for the clock.

Example:

```

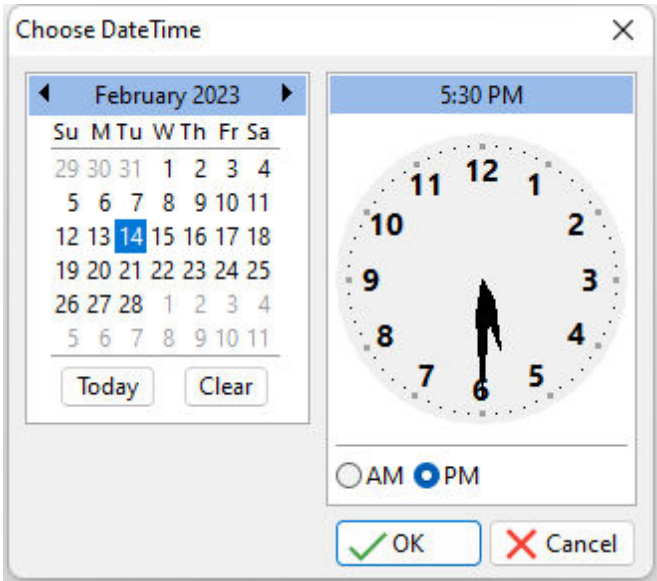
DIALOG 'Enter Time to Enter:' vEnterTime=26 vEndKey 1 +
CAPTION 'Enter Time' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for Clock' +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE TIME +
|POPUP_CLOCK_BACK_COLOR WHITE +
|POPUP_CLOCK_FACE_COLOR LIGHT GRAY +
|POPUP_CLOCK_HANDS_COLOR BLACK +
|POPUP_CLOCK_NUMBERS_COLOR NAVY +
|POPUP_CLOCK_HOURLTICKS_COLOR NAVY +
|POPUP_CLOCK_MINUTETICKS_COLOR WHITE +
|POPUP_CLOCK_HEADER_COLOR NAVY +
|POPUP_CLOCK_HEADER_FONT_COLOR WHITE +
|POPUP_CLOCK_AMPM_FONT_COLOR NAVY +
|POPUP_CLOCK_DEFAULT_TIME 12

```



1.6.10.2.5.4 DATETIME

These parameters are specific to adding a "DATETIME" custom popup menu to the the DIALOG window. The font settings for the calendar and clock will be based upon the [input field](#) parameters: INPUT_FONT_NAME, INPUT_FONT_COLOR, INPUT_FONT_SIZE.



Parameter	Value	Description
POPUP_CALENDAR_BACK_COLOR	value	Specifies the background color of the calendar
POPUP_CALENDAR_DAYS_COLOR	value	Specifies the color of the calendar days.
POPUP_CALENDAR_FILLDAYS_COLOR	value	Specifies the color of the calendar fill days.
POPUP_CALENDAR_DAYSOFWEEK_COLOR	value	Specifies the color of the calendar days of week.
POPUP_CALENDAR_LINES_COLOR	value	Specifies the color of the calendar lines.
POPUP_CALENDAR_SELBACK_COLOR	value	Specifies the color of the selected date background.

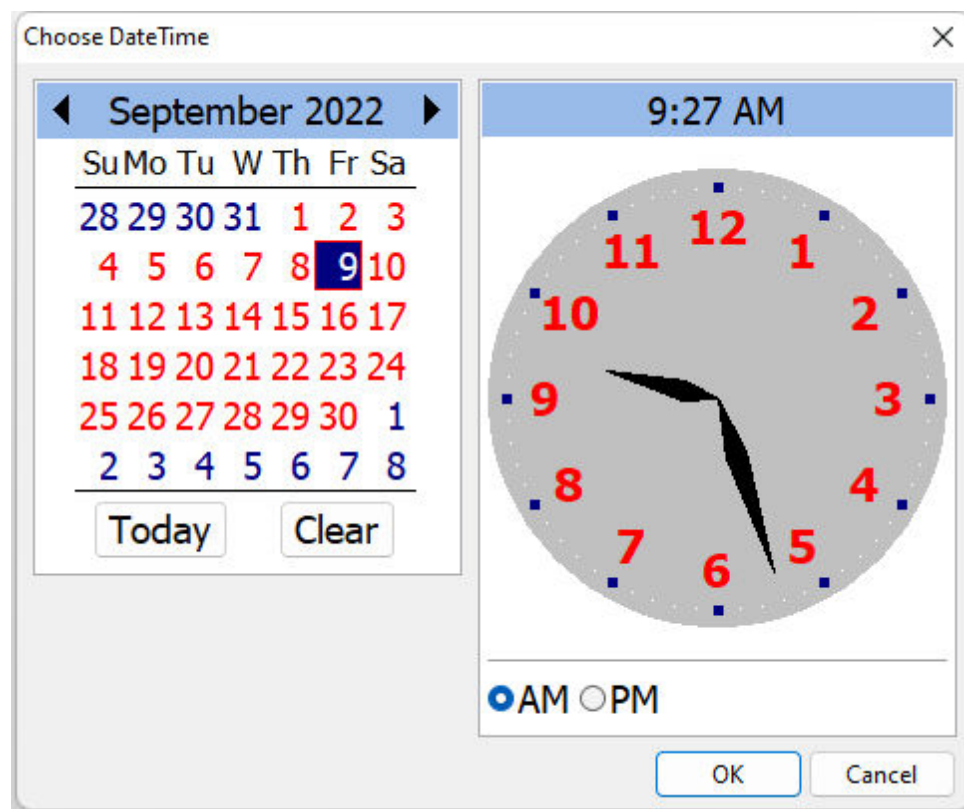
POPUP_CALENDAR_SELFORE_COLOR	value	Specifies the color of the selected date foreground.
POPUP_CALENDAR_TODAYFRAME_COLOR	value	Specifies the color of today's frame.
POPUP_CLOCK_BACK_COLOR	value	Specifies the background color of the clock.
POPUP_CLOCK_FACE_COLOR	value	Specifies the face color of the clock.
POPUP_CLOCK_HANDS_COLOR	value	Specifies the color of the clock's hands.
POPUP_CLOCK_NUMBERS_COLOR	value	Specifies the color of the clock's numbers.
POPUP_CLOCK_HOURTICKS_COLOR	value	Specifies the color of the clock's hour ticks.
POPUP_CLOCK_MINUTETICKS_COLOR	value	Specifies the color of the clock's minute ticks.
POPUP_CLOCK_HEADER_COLOR	value	Specifies the background color of the clock header.
POPUP_CLOCK_HEADER_FONT_COLOR	value	Specifies the font color of the clock header.
POPUP_CLOCK_AMPM_FONT_COLOR	value	Specifies the font color of the AM/PM radio buttons.
POPUP_CLOCK_DEFAULT_TIME	value	Specifies the default time for the clock.

Example:

```

DIALOG 'Enter Date and Time:' vDateTime=26 vEndKey 1 +
CAPTION 'Enter Date' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for Calendar/Clock' +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE DATETIME +
|POPUP_CALENDAR_BACK_COLOR WHITE +
|POPUP_CALENDAR_DAYS_COLOR RED +
|POPUP_CALENDAR_FILLDAYS_COLOR NAVY +
|POPUP_CALENDAR_DAYSOFWEEK_COLOR +
|POPUP_CALENDAR_LINES_COLOR BLACK +
|POPUP_CALENDAR_SELBACK_COLOR NAVY +
|POPUP_CALENDAR_SELFORE_COLOR WHITE +
|POPUP_CALENDAR_TODAYFRAME_COLOR RED +
|POPUP_CLOCK_BACK_COLOR WHITE +
|POPUP_CLOCK_FACE_COLOR LIGHT GRAY +
|POPUP_CLOCK_HANDS_COLOR BLACK +
|POPUP_CLOCK_NUMBERS_COLOR RED +
|POPUP_CLOCK_HOURTICKS_COLOR NAVY +
|POPUP_CLOCK_MINUTETICKS_COLOR WHITE +
|POPUP_CLOCK_AMPM_FONT_COLOR NAVY +
|POPUP_CLOCK_DEFAULT_TIME 12 +
|INPUT_FONT_NAME Tahoma +
|INPUT_FONT_SIZE 14

```

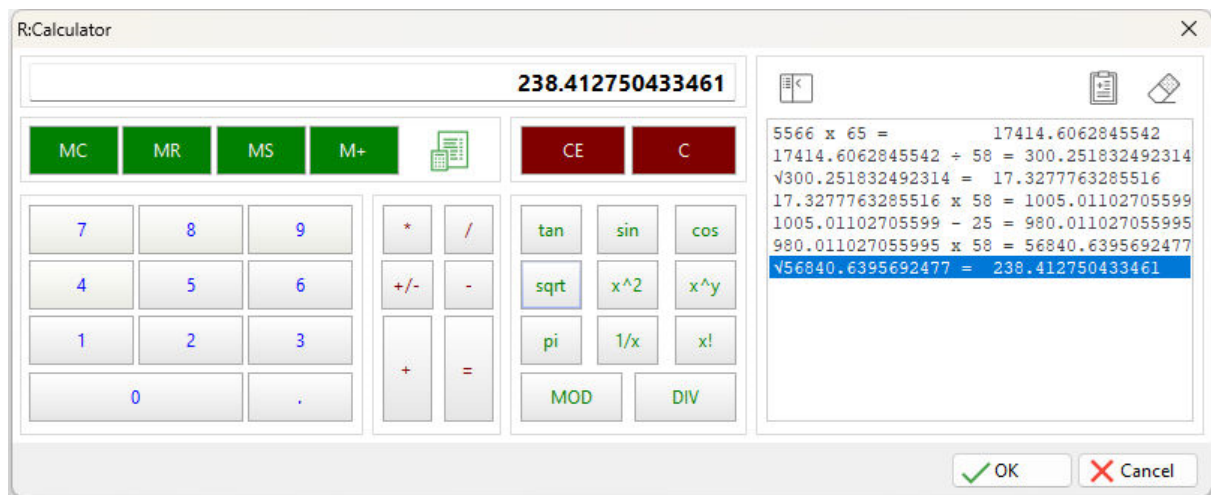


1.6.10.2.5.5 CALCULATOR

When this parameter is specific a "CALCULATOR" popup will appear for the the DIALOG window.

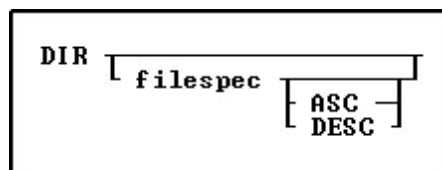
Example:

```
DIALOG 'Enter Totals for Today:' vTotals=26 vEndKey 1 +
CAPTION 'Enter Total' +
ICON APP +
OPTION DIALOG_EDIT_HINT 'Double-Click for Calculator' +
|POPUP_ENABLED TRUE +
|POPUP_DIALOG_TYPE CALCULATOR
```



1.6.11 DIR

Use the DIR command to list the files in a directory.



Option

ASC

Specifies to sort the directory output alphabetically in ascending order.

DESC

Specifies to sort the directory output alphabetically in descending order.

filespec

Specifies a filename, including the drive or directory. Wildcards can be used.

About the DIR Command

This command is similar to the operating system DIR command. The R:BASE DIR command pauses at the end of each full screen of information, based upon the current [LINES](#) setting. This command does not support the wide-display (/W) or pause (/P) switches of the operating system DIR command.

DIR lists the disk's volume label and serial number, followed followed by a list of directories and files on the disk. File information includes the file date, time, size in bytes, and name. DIR also displays the number of files (with their total bytes) and directories listed and the total bytes available on the disk. Subdirectory names, if any, are indicated with <DIR>.

If you specify a file name, R:BASE displays information for that file only. You can use an operating system wildcard character (question mark (?) or asterisk (*)) to display a group of files. The DIR command supports long (over 80 character) strings, and long strings with spaces.

Use DIR to test a file specification that uses a wildcard before using the [COPY](#), [RENAME](#), or [ERASE](#) commands. If the directory or disk is empty, or if the specified file(s) is not in the directory, DIR displays a "File not found." error message.

SHORTNAME

If you prefer the traditional format of displaying the directory contents, where the file names are listed first with the short names displayed, you can do so with the [SHORTNAME](#) setting. With SHORTNAME ON,

the DIR command output lists the file name, extension, size in bytes, and the date and time stamp. DIR also displays the number of files listed and bytes available on disk. Subdirectory names, if any, are indicated with <DIR>.

Examples

Example 1:

The following command could be used to review the RX1 database files placed in the C:\RBTI folder. The display uses SHORTNAME set to OFF.

```
R>DIR *.RX1
```

```
Volume in drive C has no label.
Volume Serial Number is 4060-5572
```

```
Directory of C:\RBTI\
```

```
11/17/2022  09:27 AM                77,492 abc1234.RX1
04/05/2022  01:56 PM                38,016 TestUse.RX1
10/03/2016  09:24 AM                76,038 USERMGMT.RX1
10/11/2022  10:46 AM                78,486 USERMGMTX5.RX1
10/12/2022  02:38 PM                80,904 USERMGMTX_WithPUBLIC.RX1
               12 File(s)                426,974 bytes
               0 Dir(s)          386,735,194,112 bytes free
```

Example 2:

The following command could be used to review the RX1 database files placed in the C:\RBTI folder. The display uses SHORTNAME set to ON.

```
R>DIR *.RX1
```

```
Volume in drive C has no label.
Directory of C:\RBTI\
```

```
abc1234  RX1      77492 11/17/2022 09:27a abc1234.RX1
TestUse  RX1      38016 04/05/2022 01:56p TestUse.RX1
USERMGMT RX1      76038 10/03/2016 09:24a USERMGMT.RX1
USERMGMT RX1      78486 10/11/2022 10:46a USERMGMTX5.RX1
USERMGMT RX1      80904 10/12/2022 02:38p USERMGMTX_WithPUBLIC.RX1
               12 File(s)                426974 bytes
               12 Dir(s)          386734391296 bytes free
```

1.6.12 DISCONNECT

Use the DISCONNECT command to write any modifications of the currently open database to disk, then close the database.

DISCONNECT

About the DISCONNECT Command

Use the DISCONNECT command before copying database files, or when you want to leave the open database without opening another database. If you keep your database on a floppy disk, use DISCONNECT to close the database before you remove the floppy disk.

The [CONNECT](#) and [EXIT](#) commands also close an open database.

If transaction processing is on when you execute DISCONNECT, R:BASE [COMMITs](#) (saves) any open transactions.

Example

The following command closes an open database:

```
DISCONNECT
```

1.6.13 DISPLAY

Use the DISPLAY command to display up to 24 lines of text that is stored in a screen file.

```
DISPLAY scrnname [ IN procfile ] [ AT scrnrow [ ,scrncol ] ]
```

Options

AT scrnrow

Specifies the top line of the display (1-24).

IN procfile

Refers to a procedure file processed through [CodeLock](#).

,scrncol

Specifies the column of the upper-left corner of the display (1-80).

scrnname

Refers to a screen block stored in a procedure file or to any disk file whose contents are to be displayed.

About the DISPLAY Command

The DISPLAY command is only supported in R:BASE for DOS.

A screen file can be a procedure, ASCII, or snapshot file. To produce a screen file that you can display with the DISPLAY command, use any of the following methods:

- The **Tools: R:BASE Editor** menu option in R:BASE for Windows.
- A word processor that creates ASCII files.

When you use the SNAP command, the snapshot file contains two screen attributes- colors and reverse video. DISPLAY interprets screen attributes stored in the snapshot file so the screen appears exactly the same as when the snapshot was taken.

When you use the **Take a Snapshot** option from the **Utilities** menu in R:BASE for Windows, the current screen is captured in bitmap form in the Windows Clipboard Viewer. This bitmap file cannot be used by the DISPLAY command.

When you display text from either an ASCII or snapshot file, the *scrname* option refers to the name of the screen file. R:BASE displays the entire contents of the screen file, which could look like the following lines.

```
This is the first line that displays on the screen
You can display up to 24 lines of text
```

When you display text that is stored in a procedure file created by CodeLock, you must use the *IN procfile* option. In this case, *scrname* refers to a screen block in the procedure file. A screen block in a procedure file looks similar to following lines. The word \$SCREEN on the first line identifies the block of text as a screen block. The second line contains the screen block name. Supply a drive letter if the screen file is not stored on the default drive, and a path if it is not in the current directory.


```
$SCREEN
scrnname
This is the first line that displays on the screen
You can display up to 24 lines of text.
```

Examples

The following command displays the screen named *screen1* in the procedure file MYPROC.APX on drive B:.

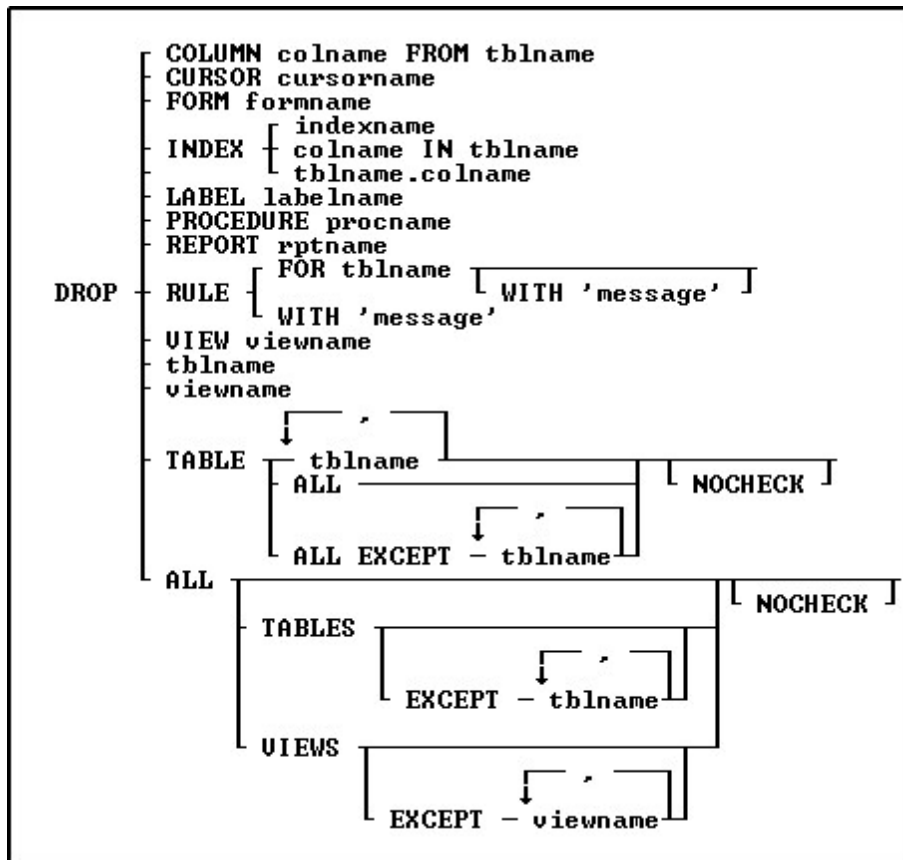
```
DISPLAY screen1 IN b:myproc.apx
```

In the following example, the first command line sets variable *v1* to a TEXT data type for text entry. The second command line displays the text that is stored in a screen file named MAINMENU in the current directory on the default drive. The third command line writes a user prompt starting at row 16, column 20 on screen. It accepts a single character as a response from the user.

```
SET VARIABLE v1 TEXT
DISPLAY mainmenu
FILLIN v1=0 USING 'Enter a letter (A-F)' AT 16,20
```

1.6.14 DROP

Use the DROP command to remove the specified item from the database.



Options

' Indicates that this part of the command is repeatable.

ALL

Removes all tables from the database.

ALL EXCEPT tblname

Removes all tables from the database except the table(s) listed.

ALL EXCEPT viewname

Removes all views from the database except the view(s) listed.

colname IN tblname

Removes the index from the specified column in the specified table.

COLUMN colname FROM tblname

Removes a column's structure and data from the specified table.

CURSOR cursorname

Removes the specified cursor. The DROP CURSOR command removes a cursor definition from memory, therefore freeing memory.

Cursors are dropped when you use the [CONNECT](#) command to open another database, or the [DISCONNECT](#) command to close the database.

FOR tblname

Removes the rule from the specified table for the specified table.

FORM formname

Removes the specified form.

INDEX

Removes the index from the column in the specified table.

indexname

Removes a named index.

LABEL labelname

Removes the specified label.

NOCHECK

Eliminates the confirmation message.

PROCEDURE procname

Removes the specified stored procedure from the database.

REPORT rptname

Removes the specified report.

RULE

Removes the specified rule.

TABLE

Removes the specified table.

tblname

Specifies the table name to be removed.

tblname.colname

Removes the index from the specified column in the specified table.

VIEW viewname

Removes the specified view.

viewname

Specifies the view name to be removed.

WITH 'message'

Removes the specified rule from all tables to which it applies. Omitting the WITH 'message' option removes all rules.

About the DROP Command

The table below lists guidelines for using the DROP command.

When you drop...	You must...
A column used in the definition of a computed column	Remove the computed column first.
A column used in a form, report, or label	Delete the column from the form, report, or label definition.
A column used in a rule	Delete the rule for that column in any table from which you removed the column.
A form, report, label, table, or view used in an application	Revise your application to reflect the changes in the database after dropping the form, report, or label.
A view or table used in a form report or label	Delete the form, report, or label from the database after dropping the form, report, or label.
A column or table used in a view	Delete the view from the database.
A table that has rules	Delete the rule with DROP if the table you are dropping is used in the WHERE clause of a rule definition. For example, a table name is used in a WHERE clause of a rule definition when a rule is used to verify a value in one table against values in another table.

R:BASE deletes the rules if the table is the table on which the rules are based.

After running the DROP command, the database item is gone, but the disk space the item occupied is not available. To recover that space, use the [PACK](#) or [RELOAD](#) commands.

You can rebuild a dropped index using [CREATE INDEX](#).

When [STATICDB](#) is set on, which activates a read-only schema mode, DROP is unavailable.

Database Access Rights with DROP

The DROP command requires that you enter the database owner's user identifier if a user identifier has been assigned with the [GRANT](#) command. However, if a user has CREATE or ALTER access rights, that user can use the DROP command to drop tables or any columns in tables to which the user has rights.

Removing Rules with DROP RULE

Before you remove a rule with the DROP RULE command, enter a [SELECT](#) command to verify that you would be removing the correct rule from the correct table(s). Use the conditions in a WHERE clause to enter the exact message and any table names that you plan to use in the DROP command. Once you have verified that the message would remove the correct rules, proceed with the DROP RULES command. For example, to verify that a DROP command with the message "Model number must be unique" would remove only the rules you want to remove, enter the following SELECT command. R:BASE would display all the rules for all the tables in the database to which this message applies.

```
SELECT * FROM SYS_RULES WHERE SYS_MESSAGE = 'Model number must be unique'
```

Examples

The following command removes the *empext* column from the *employee* table.

```
DROP COLUMN empext FROM employee
```

The following two command lines show alternative ways to remove the index from the *custid* column in the *transmaster* table.

```
DROP INDEX custid IN transmaster
DROP INDEX transmaster.custid
```

The following command removes from the database all rules with the message 'Model number must be unique.'

```
DROP RULE WITH 'Model number must be unique'
```

The following command removes any rule from the *product* table that starts with the message 'Model number.' You can use the wildcard character for MANY (%) in a message.

```
DROP RULE FOR product WITH 'Model number%'
```

The following command removes the cursor named *cursor1* from memory.

```
DROP CURSOR cursor1
```

1.6.15 DUPROW

Use the DUPROW command to duplicate the current row of data currently being edited or entered in a form.

DUPROW

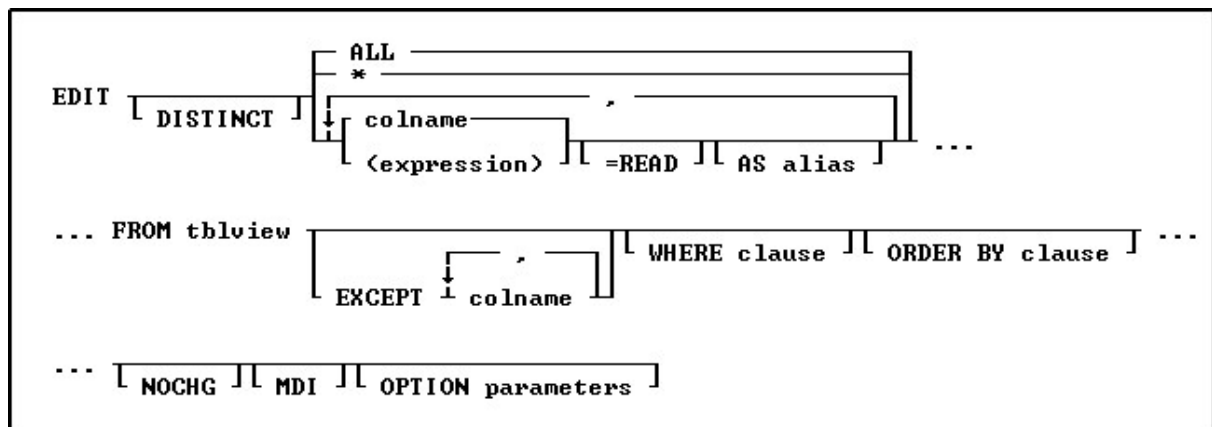
About the DUPROW Command

Use the DUPROW command to duplicate the current row of data currently being edited or entered in a form.

1.7 E

1.7.1 EDIT

Use the EDIT command to change data in the Data Editor/Browser.



Options

*
Specifies all columns.

'
Indicates that this part of the command is repeatable.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

ALL

Specifies all columns.

AS alias

Determines the alias of the column.

colname

Specifies a column name. The column name is limited to 128 characters.

DISTINCT

Suppresses the display of duplicate rows. If you specify DISTINCT, you cannot change the data that is displayed.

EXCEPT colname

Specifies column names to be excluded from the Data Browser. The EXCEPT clause is helpful with tables which have many columns, to exclude columns from the result set. EXCEPT saves time to avoid typing a long list of column names. When using EXCEPT there can only be one table specified.

FROM tbleview

Specifies the table.

MDI

Specifies a modeless Data Editor window, where other windows within R:BASE can be accessed without closing the current Data Editor window first. This option is used in R:BASE for Windows only.

NOCHG

Prevents changes to the issued EDIT command and stops users from reaching additional data by adding columns to the command or changing the command's WHERE clause with the Update Query menu option. NOCHG is particularly useful when the BROWSE command is issued within an application or command file.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

=READ

Specifies that the column is read-only, preventing changes to data.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

OPTION parameters - provides the ability customize the Data Editor Window. These enhanced parameters are supported at the R> Prompt, in command files, EEPs and stored procedures.

OPTION Parameters	Value	Description
CAPTION	value	specifies the window caption
CELL_HINTS	ON (Default) OFF	specifies if NOTE column cell hints are displayed
COLUMN_WIDTH[x] y	values	specifies the column width for a column, where X is the zero-based column index and Y is the width in pixels
CURRENT_FIELD_VALUE_VAR	value	specifies to capture the field value for the current focused cell
CURRENT_FIELD_VAR	value	specifies to capture the column name for the current focused cell
EXPORTRESULT	ON (Default) OFF	specifies if the "Export Result to->" menu item is available
GRIDVIEW		specifies the "View as Grid" mode is displayed
HEIGHT	value	specifies the window height, in pixels

IMAGE_HINT	ON (Default) OFF	specifies if image hints are displayed
LEFT	value	specifies the window location, in pixels, from the left of the screen over
LOCK	nnn	specifies to lock a number of columns
MODAL	ON (Default) OFF	specifies if the window is modal or non-modal. With the MODAL set to OFF, the Data Editor window displays as a separate button on the Windows task bar.
NO_FOCUS_FIELDS	value	specifies columns that cannot receive focus. The value is a comma separated list of columns.
ON_CELL_CLICK_EEP	value	specifies to execute the EEP command file when a cell is clicked
ON_CELL_DBL_CLICK_EEP	value	specifies to execute the EEP command file when a cell is double-clicked
PRINTDATA	ON (Default) OFF	specifies if the "Print Data" menu item is available
READONLY	ON OFF	specifies if the data is read only
ROW_NUMBERS	ON OFF (Default)	specifies if a record number is displayed (column to the far left)
ROWVIEW		specifies the "View as Row" mode is displayed
ROW_VIEW_DESCRIPTIONS	ON OFF (Default)	specifies to display the column comments from the table definition (when present) to the right of the data, when ROWVIEW mode is specified
SAVERESULT	ON (Default) OFF	specifies if the "Save Data As->" menu item is available
SAVERESULTTEMPONLY	ON OFF (Default)	specifies if the "Save Results As->" menu items are available for both permanent and temporary tables/views
TOP	value	specifies the window location, in pixels, from the top of the screen down
TREEGRIDVIEW		specifies the "View as Tree Grid" mode is displayed
UPDATE_QUERY	ON (Default) OFF	specifies if the query can be updated
WIDTH	value	specifies the window width, in pixels
WINDOW_STATE	MAXIMIZED NORMAL MINIMIZED	specifies the windows state
ZEBRA_STRIPE	ON OFF	specifies if zebra stripes are displayed

About the EDIT Command

The EDIT command starts the Data Editor and displays the data you specify. In a multi-user environment, R:BASE checks whether the row of data you are editing has been changed by another user and notifies you if it has - this is called concurrency control.

You can use the EDIT command to modify data when the following conditions are met:

- You have not used the DISTINCT option in the command.
- You have rights to change the data in the specified table. Access rights are assigned with the [GRANT](#) command.

Notes:

- The EDIT command will display the data within a multi-table view (a view based on multiple tables), but the data will not be editable.
- A view with a [GROUP BY](#) parameter is also not editable.

Examples

Example 01:

The following command displays all columns in the *transmastertable* and orders the rows by the customer identification numbers contained in the *custid* column.

```
EDIT * FROM transmaster ORDER BY custid
```

Example 02:

The command below displays the following columns from the *customertable*: *custid*, *company*, *custaddress*, *custcity*, *custstate*, and *custzip*. The only rows that are displayed are rows that have 100 for a customer identification. The *custid* and *company* columns are specified to be read-only and cannot be edited.

```
EDIT custid=READ, company=READ, custaddress, custcity, +  
custstate, custzip FROM customer WHERE custid = 100
```

Example 03:

The following command open all records for the Customer table with a maximized Data Editor window and "Customer Data" as the caption.

```
EDIT ALL FROM customer OPTION WINDOW_STATE MAXIMIZED|CAPTION 'Customer Data'
```

Example 04:

The following will start the Data Editor for *custid* 125, displaying the data as a single row in read only mode.

```
EDIT ALL FROM customer WHERE custid = 125 OPTION WINDOW_STATE NORMAL|ROWVIEW|  
READONLY ON
```

Example 05:

The following opens the Data Editor for the *Customer* table, with a limited height and width, and is located near the top left corner of the screen.

```
EDIT ALL FROM Customer +  
OPTION WINDOW_STATE NORMAL +  
|TOP 25 +  
|LEFT 25 +  
|WIDTH 1000 +  
|HEIGHT 500
```

Example 06:

The following opens the Data Editor for active customers in the *Customer* table, with column alias names, a custom caption, and specified column widths for three that contain shorter data lengths.

```
EDIT +  
CustID AS `Customer ID`, +  
Company AS `Company Entity`, +  
CustAddress AS `Home Office Address`, +  
CustCity AS `Home Office City`, +  
CustState AS `Home Office State`, +  
CustZip AS `Home Office Zip Code`, +  
CustPhone AS `Primary Phone Number`, +  
CustFax AS `Fax Number`, +  
LastUpdateDate AS `Last Date Modified`, +  
LastOrderDate AS `Last Order Date`, +  
CustURL AS `Company Home Page`, +
```

```

CustEmail AS `Primary Email Address`, +
CustStatus AS `Active Status`, +
PaymentTerm AS `Payment Terms`, +
CreditLimit AS `Imposed Credit Limit` +
FROM Customer +
WHERE CustStatus = 'A' +
ORDER BY Company ASC +
OPTION CAPTION 'Active Customer Information'+
|WINDOW_STATE NORMAL +
|COLUMN_WIDTH[4] 50 +
|COLUMN_WIDTH[5] 50 +
|COLUMN_WIDTH[12] 50

```

Example 07:

The following opens the Data Editor for contacts with email addresses in the *Contacts* table, and launched the CellDbClick.rmd command file when a cell is double clicked.

```

EDIT CustID,ContFName,ContLName,ContEmail FROM Contact +
WHERE ContEmail IS NOT NULL +
OPTION ON_CELL_DBL_CLICK_EEP CellDbClick.rmd

```

Example 08:

The following opens the Data Editor for California customers in the *Customer* table, and captures the column name and value wherever the focus is upon a cell.

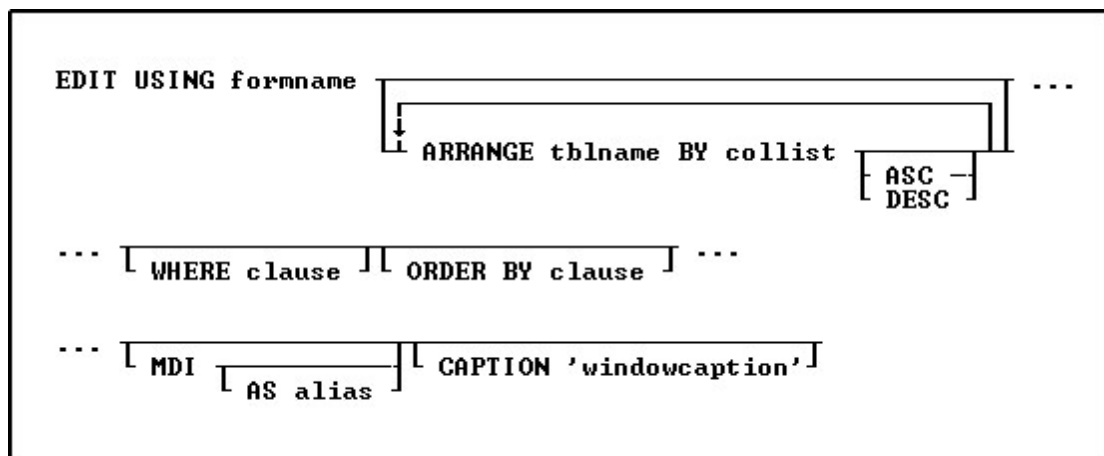
```

SET VAR vCurFieldName TEXT = NULL
SET VAR vCurFieldValue TEXT = NULL
EDIT CustID,Company,CustCity,CustState,CustZip FROM Customer +
WHERE CustState = 'CA' +
OPTION CURRENT_FIELD_VAR vCurFieldName +
|CURRENT_FIELD_VALUE_VAR vCurFieldValue

```

1.7.2 EDIT USING

Use the EDIT USING command to display a form with which to view, add, change, or delete data.



Options

formname

Specifies a form name. The form name is limited to 128 characters.

ARRANGE tblname BY collist

Sorts the rows displayed by a form's lower-level tables. Each lower-level table in the form that you want to sort requires its own ARRANGE clause. Now you can specify up to twenty-five tables to ARRANGE and up to five columns in each table on which R:BASE will sort.

**ASC
DESC**

Specifies whether to sort a column in ascending or descending order.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

MDI

This option is used in R:BASE for Windows only; specifies a modeless form-a modeless form allows you to access other windows without closing the form first.

AS alias

This option is used in R:BASE for Windows only. Allows the user to specify a name for this instance of the form. This option is used in conjunction with the MDI option so that the user can programmatically specify the form with the [SETFOCUS](#) command.

CAPTION 'windowcaption'

This option is used in R:BASE for Windows only. Specifies a caption for the window title bar.

Examples

```
EDIT USING formname CAPTION 'Form Caption'
```

```
EDIT USING formname MDI
```

About the EDIT USING Command

The EDIT USING command displays data in a previously created form. When the form was created, the user who created the form decided which database actions could be used on the form's tables. These database actions appear on the **Forms** menu.

In a multi-table form, the WHERE and ORDER BY clauses apply only to the first table in the form.

The EDIT USING command allows a form that includes a form called from an entry/exit procedure (EEP). The number of times a form can be invoked within another form is limited only by the amount of stack space available.

In R:BASE for Windows EDIT USING *formname* MDI AS *alias* can be used to start a form window that can later be accessed with a [SETFOCUS](#) command.

Notes:

- The EDIT USING command will display the data in a form based upon a multi-table View (a View based on multiple tables), but the data will not be editable.
- A View with a [GROUP BY](#) parameter is also not editable.

Examples

The following command displays the form named *CustomerEdit* with the rows from the first table, ordered by customer identification.

```
EDIT USING CustomerEdit ORDER BY CustID
```

The command below displays the form named *CustomerEdit* with the rows from the first table that have a customer identification number equal to 100.

```
EDIT USING CustomerEdit WHERE CustID = 100
```

The RRBYW20 database contains a five-table form called *SalesTransactions*. The command below shows how the form's four subordinate tables, *InvoiceHeader*, *InvoiceDetail*, *Employee* and *Contact*, can be sorted using an **ARRANGE** clause. The **ARRANGE** clause sorts the second table, *InvoiceHeader*, by the *TransID* column, in ascending order, sorts the third table, *InvoiceDetail*, by the *DetailNum* column, in ascending order, sorts the fourth table, *Employee*, by *EmpLName* and *EmpFName*, in ascending order and the fifth table, *Contact*, by the *ContLName* and *ContFName*, in ascending order.

```
CONNECT RRBYW20
```

```
EDIT USING SalesTransactions +
ARRANGE InvoiceHeader BY TransID ASC, +
ARRANGE InvoiceDetail BY DetailNum ASC, +
ARRANGE Employee BY EmpLName ASC, EmpFName ASC, +
ARRANGE Contact BY ContLName ASC, ContFName ASC +
ORDER BY CustID
```

1.7.3 ENTER

Use the **ENTER** command to use a form to add information to tables.

```
ENTER [ USING ] formname [ FOR n ROWS ] ...
... [ MDI [ AS alias ] ] [ CAPTION 'windowcaption' ]
```

Options

USING

Specifies a modal form - modal means that you must close the form before you can click in another window. If you do not specify an option, a modal form is the default.

formname

Specifies a form name. The form name is limited to 128 characters.

FOR n ROWS

Limits to *n* (an integer) the number of rows that can be entered into the form's first table during one session of data entry.

MDI

Specifies a modeless form - a modeless form allows you to access other windows without closing the form first. If you do not specify an option, a modal form is the default.

AS alias

Allows the user to specify a name for this instance of the form. This option is used in conjunction with the **MDI** option so that the user can specify the form with the [SETFOCUS](#) command.

CAPTION 'windowcaption'

Specifies a caption for the window title bar.

About the ENTER Command

This command displays a form for data entry. When the form was created, the creator of the form specified which database actions the user could use on the form's tables. These database actions appear on the **Forms** menu.

Notes:

- The ENTER command will not enter data in a form based upon a multi-table view (a view based on multiple tables), as the data is not editable.
- A view with a [GROUP BY](#) parameter is also not editable.

Examples

The following command displays the *transform* form and allows users to enter one row of data to the first table served by *transform*. You can enter as many rows of data in subsequent tables as needed for the one row entered in the first table. After you enter the specified number of rows, R:BASE returns you to the R> Prompt, or the next line in the command file. Specifying the number of rows of data to enter is useful in applications that require other actions to take place after loading each entry.

```
ENTER transform FOR 1 ROW
```

1.7.4 ERASE

Use the ERASE command to delete a file.

ERASE filespec

Options

filespec

Specifies a file name with a drive and path specification in the form D:\PATHNAME\FILENAME.EXT.

About the ERASE Command

If the file is in the current directory, you can omit its path name in the file specification.

R:BASE requires you to close a database before erasing it. ERASE is irreversible. When you use a wildcard with ERASE at the R> Prompt, R:BASE prompts you to press [Esc] to cancel, or [Enter] to delete the files.

Examples

The following command erases all files in the OLDSALES directory on the current drive.

```
ERASE oldsales\*.*
```

The following command lines close then erase the four files of the *mydb* database in the OLDDATA directory on drive E:.

```
DISCONNECT  
ERASE e:\olddata\mydb.rb?
```

1.7.5 EXIT

Use the EXIT command to leave R:BASE and return to the operating system. EXIT closes an open database, releases memory taken by R:BASE, and releases file handles.

EXIT

Notes:

- When exiting an application it is advised to [DISCONNECT](#) from the database first.
- Do not use EXIT within an [IF...ENDIF](#) control structure. Instead, use a [GOTO](#) in order to EXIT an application outside of the IF...ENDIF.

1.7.6 EXITFORM

Use the EXITFORM command to exit the current form currently being edited or entered. The EXITFORM command was replaced with the [CLOSEWINDOW](#) command.

EXITFORM

1.7.7 EXPLAIN

Use the EXPLAIN command to show the decision the optimizer made when executing a command.

EXPLAIN command

The EXPLAIN keyword must be followed by a command, which will output the optimizer selections for the query. Use EXPLAIN to determine if you need to use other optimization methods, such as [MANOPT](#). You can see the differences when different indexes or table orders are specified. For example:

```
R>EXPLAIN SELECT * FROM ProdView
R> prompt
01/26/2023 01:54 PM
Query 1=* FROM ProdView"
Query 2=SELECT
T1.TransDate,T2.TransID,T2.DetailNum,T2.Model,T2.Units,T2.Price,T2.Discount,T2.SalePri
ce,T2.ExtPrice,T3.CustID,T3.Company,T3.CustAddress,T3.CustCity,T3.CustState,T3.CustZip
,T3.CustPhone,T3.ModLevel,T3.LastUpdateDate,T3.LastUpdateTime,T3.LastOrderDate,T3.Cust
URL,T3.CustEMail,T3.CustStatus,T3.PaymentTerm,T3.CreditLimit,T4.EmpID,T4.EmpFName,T4.E
mpLName,T5.ProdName FROM InvoiceHeader T1,InvoiceDetail T2,Customer T3,Employee
T4,Product T5 WHERE (((T1.TransID = T2.TransID) AND (T1.CustID = T3.CustID ) AND
(T1.EmpID = T4.EmpID) AND (T2.Model = T5.Model)))"
Table Order 1=ProdView
SelectCost=1. (OptimizationTime=0ms)
  ProdView Sequential

Table Order
2=InvoiceHeader,Customer.CustID,Employee.EmpID,InvoiceDetail.TransID,Product.Model
SelectCost=1.439 (OptimizationTime=1ms)
  InvoiceHeader Sequential
  Customer (ColumnName=CustID,Type=P) Random Dup=1. Adj=1.
  Employee (ColumnName=EmpID,Type=P) Random Dup=1. Adj=1.
  InvoiceDetail (ColumnName=TransID,Type=F) Random Dup=1.525773 Adj=1.
  Product (ColumnName=Model,Type=P) Random Dup=1. Adj=1.
```

The EXPLAIN command shows the optimizer selections for both implicit and explicit [SELECT](#)s. In the above, the data from the *ProdView* view is retrieved sequentially, but the join used to create the data for

ProdView uses indexes. The optimizer table order and index use matches that returned by [#TABLEORDER](#).

The EXPLAIN output includes the command file name and next byte offset (location of command within the file) and the current date and time. Commands from the R> Prompt are so noted. The query and table order are listed. The values for "Dup" and "Adj" are the Duplicate Factor and Adjacency Factor values, which are available for review in the Data Designer within the Keys/Indexes area.

About the EXPLAIN command

EXPLAIN lets you easily test query options and see the differences. Every query made by R:BASE sends the optimizer selections to the current output.

The output displays the optimization time in milliseconds (OptimizationTime=30ms). This time is how long it took R:BASE to determine the best table and index order. Each table order and index the optimizer checks is given a value (SelectCost=4.84453). The lowest value or select cost determines which query order is used. There are four types of access methods in the EXPLAIN output that can be used to retrieve the data:

Sequential	No index was used. The rows are retrieved from the table in the order they were entered.
Random	An index access method.
VeryRandom	An index access method.
IndexOnly	No data was retrieved from file 2, all the data was retrieved from file 3.

When index access is used (IndexOnly, Random, VeryRandom), the column name or index name and the index type are displayed. For example:

```
TransMaster (ColumnName=EmpID, Type=F) IndexOnly Dup=3.8 Adj=1
TransMaster (IndexName=DateIndex,Type=I) Random Dup=1.055556 Adj=1
```

The index types are:

P	Primary Key
F	Foreign Key
I	Index

The EXPLAIN command was introduced to simplify the use of the MICRORIM_EXPLAIN system variable.

Examples:

Example 01 (Display the optimizer results for the TCrossTabView view):

```
R>EXPLAIN SELECT * FROM TCrossTabView
R> prompt
01/30/2023 11:27 AM
Query 1=* FROM TCrossTabView"
Query 2=SELECT T1.EmpLName,T2.Company,T4.ExtPrice,T5.ProdName FROM Employee
T1,Customer T2,InvoiceHeader T3,InvoiceDetail T4,Product T5 WHERE T3.EmpID = T1.EmpID
AND T3.CustID = T2.CustID AND T4.TransID = T3.TransID AND T5.Model = T4.Model"
Table Order 1=TCrossTabView
SelectCost=1. (OptimizationTime=0ms)
    TCrossTabView Sequential

Table Order
2=InvoiceHeader,Employee.EmpID,Customer.CustID,InvoiceDetail.TransID,Product.Model
SelectCost=1.439 (OptimizationTime=0ms)
    InvoiceHeader Sequential
    Employee (ColumnName=EmpID,Type=P) Random Dup=1. Adj=1.
    Customer (ColumnName=CustID,Type=P) Random Dup=1. Adj=1.
    InvoiceDetail (ColumnName=TransID,Type=F) Random Dup=1.525773 Adj=1.
```

Product (ColumnName=Model,Type=P) Random Dup=1. Adj=1.

Example 02 (Display the optimizer results for the InvoicesMaster view):

```
R>EXPLAIN SELECT * FROM InvoicesMaster
R> prompt
01/26/2023 02:13 PM
Query 1=* FROM InvoicesMaster"
Query 2=SELECT T1.TransID AS TransID,T1.CustID AS CustID,T1.EmpID AS
EmpID,T1.TransDate AS TransDate,T1.BillToCompany AS BillToCompany,(T1.BillToAddress +
', ' + T1.BillToCity + ' ' + T1.BillToState + ' ' + T1.BillToZip) AS
BillToAddress,T1.BillToState AS BillToState,T1.BillToZip AS BillToZip,T1.ShipToCompany
AS ShipToCompany,(T1.ShipToAddress + ', ' + T1.ShipToCity + ' ' + T1.ShipToState + ' '
+ T1.ShipToZip) AS ShipToAddress,T1.ShipToState AS ShipToState,T1.ShipToZip AS
ShipToZip,T1.NetAmount AS NetAmount,T1.Freight AS Freight,T1.Tax AS
Tax,T1.InvoiceTotal AS InvoiceTotal,IFEQ (T2.DetailNum,1,1,0) AS
InvoiceCounter,T2.DetailNum AS DetailNum,T2.Model AS Model,T2.Units AS Units,T2.Price
AS Price,T2.Discount AS Discount,T2.SalePrice AS SalePrice,T2.ExtPrice AS
ExtPrice,T3.ProdName AS ProdName,T3.ProdDesc AS ProdDesc,T3.ListPrice AS ListPrice,
(T4.EmpFName + ' ' + T4.EmpLName) AS EmpName,T4.EmpPhone AS EmpPhone,T4.EmpExt AS
EmpExt,T4.EMailAddress AS EMailAddress,T4.EmployeePhoto AS EmployeePhoto FROM
InvoiceHeader T1,InvoiceDetail T2,Product T3,Employee T4 WHERE T1.TransID = T2.TransID AND
T2.Model = T3.Model AND T1.EmpID = T4.EmpID ORDER BY T1.BillToCompany,T1.TransID
ASC,T2.DetailNum ASC"
Table Order 1=InvoicesMaster
SelectCost=1. (OptimizationTime=0ms)
    InvoicesMaster Sequential

Table Order 2=InvoiceHeader,InvoiceDetail.TransID,Product.Model,Employee.EmpID
SelectCost=1.393 (OptimizationTime=0ms)
    InvoiceHeader Sequential
    InvoiceDetail (ColumnName=TransID,Type=F) Random Dup=1.525773 Adj=1.
    Product (ColumnName=Model,Type=P) Random Dup=1. Adj=1.
    Employee (ColumnName=EmpID,Type=P) Random Dup=1. Adj=1.
```

1.8 F

1.8.1 FCONVERT

The FCONVERT command is used to perform file conversions between ANSI, UTF-8, UTF-8 BOM, Unicode, and Base64 files.

```
FCONVERT oldtype oldfilename TO newtype newfilename [ NONE ]
```

Options

newfilename

Specifies the new file end result

newtype

Specifies the file type, which can be ANSI, UTF8, UNICODE, or BASE64

NONE

Specifies to encode the new file as UTF-8 without BOM.

oldfilename
Specifies the name of the existing file to convert (encode/decode)

oldtype
Specifies the file type, which can be ANSI, UTF8, UNICODE, or BASE64

Note
While ANSI, UTF-8, and Unicode refer to styles of text file encoding, users can convert any file type to Base64 and from it.

Examples

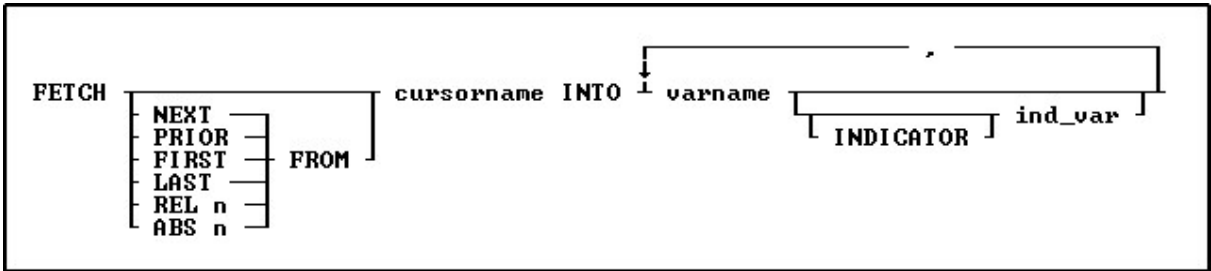
```
FCONVERT ANSI Transactions_ANSI.ASC TO UNICODE Transactions_Unicode.ASC

FCONVERT ANSI Quotes_ANSI.ASC TO UTF8 Quotes_UTF8.ASC NONE

FCONVERT BASE64 Order_BASE64.ASC TO ANSI Order_ANSI.ASC
```

1.8.2 **FETCH**

Use the FETCH command to position the cursor on a row specified by the [DECLARE CURSOR](#) command, and place values from the columns into global variables.



Options

ABS *n*
The value *n* is the *n*th row in the cursor list. The current cursor location is not relevant. Positive numbers count from the first row in the list. Negative numbers force an end-of-data condition. This option applies only to scrolling cursors, which are defined with the DECLARE CURSOR command.

cursorname INTO
Names the cursor from which to fetch data into the specified variable list.

FIRST
Specifies the first row in the cursor list. This option applies only to scrolling cursors, which are defined with the DECLARE CURSOR command.

INDICATOR *ind_var*
Stores the status of the variable: non-null (0) or null (-1).

LAST
Specifies the last row in the cursor list. This option applies only to scrolling cursors, which are defined with the DECLARE CURSOR command.

NEXT
Specifies the next entry the cursor points to. This option applies only to scrolling cursors, which are defined with the DECLARE CURSOR command.

PRIOR
Specifies the prior entry the cursor points to. This option applies only to scrolling cursors, which are defined with the DECLARE CURSOR command.

REL *n*

Moves the cursor *n* rows. Positive integers move forward, and negative integers move backwards. For example, if *n* is 5, the cursor moves forward 5 rows. This option applies only to scrolling cursors, which are defined with the [DECLARE CURSOR](#) command.

varname

Specifies a variable name, which must be unique among the variable names within the database. The variable name is limited to 128 characters.

About the FETCH Command

FETCH moves the cursor to the next available row referred to by the DECLARE CURSOR command and also accommodates scrollable cursors specified by DECLARE CURSOR. FETCH retrieves the values of columns in the order in which the columns were specified by DECLARE CURSOR. The [LIST CURSORS](#) command lists all the defined cursors.

FETCH *cursorname* without any variable specification will retrieve the next row from the cursor. Use the [SET VAR](#) *varname* WHERE CURRENT OF *cursorname* to retrieve the columns you need.

Using the Sqlcode Variable

You must check the *sqlcode* variable with each use of FETCH to verify that all rows specified by DECLARE CURSOR have been found. Use the WHENEVER *sqlcode* command to check for SQL processing errors other than data-not-found errors.

Use [WHENEVER NOT FOUND](#) to check for a data-not-found errors. When you use the WHENEVER NOT FOUND command, data-not-found error checking is automatic; however, you must use the [LABEL](#) command. When a data-not-found error occurs, control passes to the command line specified by the LABEL command and the subsequent error-handling commands.

Using Indicator Variables

If the data contains null values, use indicator variables to capture the status of a value. If you do not use indicator variables, R:BASE displays an error message when it encounters a null value, but produces no rows.

Placing a Value into a Numeric Variable

If you use FETCH to place a value into a variable that has not been previously defined and has a NUMERIC data type, then that variable acquires the precision and scale of the column from which the value is fetched.

Using the FETCH Command Without Variable Specification

Using FETCH *cursorname* without any variable specification will retrieve the next row from the cursor. Use the [SET VAR](#) *varname* WHERE CURRENT OF *cursorname* to retrieve the columns you need.

Example

The following command lines fetch every other row from a table.

```
DROP CURSOR C1
DECLARE c1 SCROLL CURSOR FOR SELECT transid, transdate, +
    netamount FROM transmaster ORDER BY netamount DESC
OPEN c1
FETCH c1 INTO vtransid ind1, vtransdate ind2, vnetamount ind3
SELECT COUNT(*) INTO vtotcount i1 FROM transmaster
SET VAR vcount INT=0
WHILE sqlcode <> 100 THEN
    SET VAR vcount = (.vcount+1)
    WRITE 'Total count', .vtotcount, 'Cursor count', .vcount
    --fetch every other row
```



```

    FETCH REL 2 FROM c1 INTO vtransid int1, +
        vtransdate ind2, vnetamount ind3
ENDWHILE
CLOSE C1
DROP CURSOR C1

```

1.8.3 FILLIN

Use the FILLIN command to prompt the user to enter a value at the keyboard.

FILLIN <i>varname</i> [<i>=width</i>] [USING ' <i>message</i> ']

Options

=width

Specifies the width of the entry field. If set to zero, you do not need to press [Enter] after the response. A single key entry is accepted, although not displayed on the screen. When *varname*=0 is specified, you can capture these keystrokes: [Enter], [Esc], [Home], [End], [PgUp], [PgDn], [Up Arrow], [Down Arrow], [Tab], and [Shift]+[Tab]. For example, [PgDn] is recorded as [PgDn] in the variable. The [Down Arrow] key is recorded as [Down] and [Up Arrow] as [Up]. The variable used with =0 is always a TEXT data type. Alternatively, you can omit this option and use the LASTKEY function to capture the last key pressed. (This is useful with [EDIT](#).) If a height and length are specified, the width references the width of the FILLIN box.

USING 'message'

Specifies the message text (a 1- to 80-character text string in the currently defined QUOTES character) to be displayed. You can use a dot variable in place of it. For example, if you enter FILLIN *var1* USING *.var2*, the screen prompt is the contents of *variable2*.

varname

Specifies a variable name, which must be unique among the variable names within the database. The variable name is limited to 128 characters.

About the FILLIN Command

The value entered at the keyboard is stored in a global variable.

Note: It is recommended to replace code instances of the FILLIN command with the [DIALOG](#) command, which has many additional display options and is more user friendly.

Setting the Data Type

If the data type of the value that is entered is unambiguous, then the variable is set to that data type. For example, if you enter 123, the variable is given an INTEGER data type. If you wanted the data type to be TEXT, you would have to set the variable's data type. If the data type is questionable—for example, when entering a date—be sure to define the variable's data type. Also, if you use =0 to accept a single keystroke, the variable's data type must be TEXT.

Examples

The following command lines ensure that the data type is TEXT. The quotation marks around the message are not displayed on screen.

```

SET VARIABLE vfill TEXT
FILLIN vfill=10 USING 'Enter a number:'

```

The following command prompts the user for a database name and stores it in the variable v2. The database name could then be used in a [CONNECT](#) command, for example, CONNECT *.v2*. The dot before the variable indicates that R:BASE is to use the current value of the variable.

```
FILLIN v2 USING 'Enter the name of the database you want to open: '
```

In the following command lines, the first command line sets variable *v1* to TEXT and *A* as its value. The *=1* in the FILLIN command lets the user enter a single character or accept the letter *A* by pressing [Enter]. After the user responds, the program stores the response in the variable *v1* and continues to the next line in the command file.

```
SET VARIABLE v1 TEXT = 'A'
FILLIN v1=1 USING 'Enter a letter (A-F): '
```

The following command lines use the *=0* option to allow a single keystroke entry, then check the variable for [PgUp] and [PgDn]. If the response is [PgUp], then *vend* contains the value [PgUp] and [GOTO](#) transfers processing to the lines following [LABEL top](#). If the response is [PgDn], control is passed to the lines following [LABEL bottom](#). If neither [PgUp] or [PgDn] is entered, control passes to [LABEL tryagain](#) and the FILLIN is repeated.

```
LABEL top
.
.
.
LABEL tryagain
FILLIN vend=0 USING 'Press PgUp to start again, PgDn to continue.'
SWITCH (.vend)
CASE '[PgUp]'
    GOTO top
CASE '[PgDn]'
    GOTO bottom
DEFAULT
    GOTO tryagain
ENDSW
.
.
.
LABEL bottom
```

The following example creates a FILLIN box 20 characters wide by three lines deep that can contain a maximum of 100 characters. If you designate a box to contain more characters than the size of the box allows, you automatically create a scrolling box.

```
FILLIN v1=20,3,100 USING 'Notation:'
```

1.8.4 FIRST

Use the FIRST command in an entry/exit procedure (EEP) to move to the first row.

FIRST

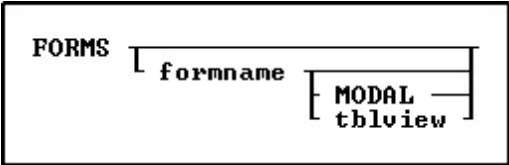
1.8.5 FLUSH

Use the FLUSH command to force database information to be written to disk.

FLUSH

1.8.6 FORMS

Use the FORMS command to start the Form Designer.



Options

formname

Names an existing form to modify. The form name is limited to 128 characters.

MODAL

Launches a form in the designer from a command block, command file, or EEP. The Form Designer must be closed in order to continue to the next command.

tblview

This option is for R:BASE for DOS only, when you are creating a new form. This option is available only if you start to build a form from the R> Prompt with a database open. Specify both *formname* and *tblview* when you want to build a new form. R:BASE displays the Form Designer screen where you can begin to define the new form.

Example

The following command launches the *Contacts* form within the Form Designer.

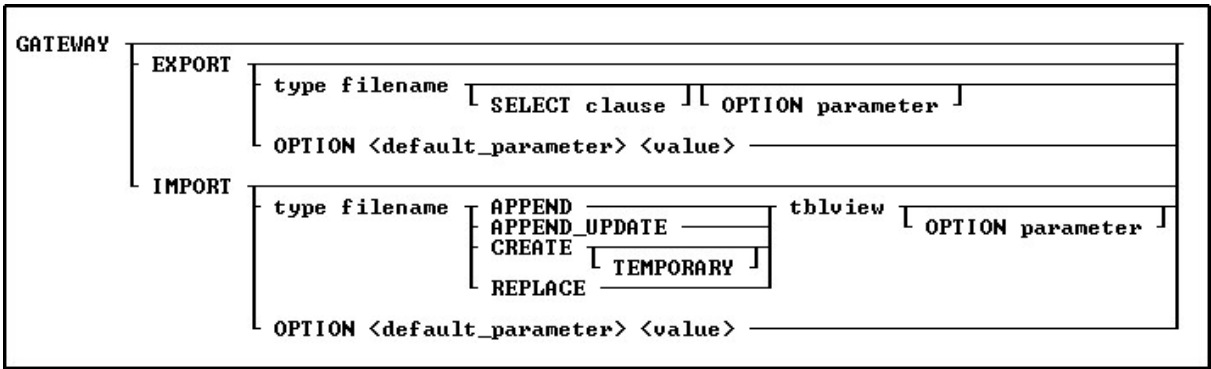
FORMS Contacts

1.9 G

1.9.1 GATEWAY

Use the GATEWAY command to start the import/export of data. Using the **GATEWAY** command at the R> Prompt or within command files you can automate the entire task of providing desired data formats to your end-users and/or clients. There are NO limits on importing or exporting numerous records into and out of R:BASE for Windows databases with the GATEWAY command.

IMPORTANT NOTE ABOUT GATEWAY: You MUST be aware of both your source file structure and table structure for the command to function properly. The table and file structure MUST coincide with each other.



EXPORT

Specifies to export an R:BASE table or view to an external file.

EXPORT Dialog

Specifies to launch the File Gateway, displaying only the "Export" options, with or with default values specified.

IMPORT

Specifies to import an external file to an R:BASE table.

IMPORT Dialog

Specifies to launch the File Gateway, displaying only the "Import" options, with or with default values specified.

OPTION <default_parameter> <value>

Specifies to launch the File Gateway with the preferred parameters and valued predefined.

About the GATEWAY Command

Use the Import/Export utility to import a file from another program into a new or existing table in an R:BASE database or export data from an R:BASE database into a file that can be imported into another program or R:BASE database.

Users can access File Gateway user interface module either by clicking on "File Gateway" icon on the R:BASE Main Toolbar or typing **GATEWAY** at the R> Prompt or in a command file. The File Gateway interface may be launched with default options specified, using parameters for exporting and importing data.

The following types of files are supported to import or export data:

- Import Formats include:

- dBase (*.dbf)
- Paradox (*.db)
- Text (*.txt)
- Comma Delimited (*.csv)
- Text Data (*.dat)
- Text Printout (*.prn)
- Tab Delimited (*.tab)
- Tilde Delimited (*.til)
- ASCII (*.asc)
- HTML (*.htm, *.html)
- MS Excel Spreadsheet (*.xls)
- MS Excel Workbook (*.xls)
- Lotus 1-2-3 (*.wk1)
- Lotus 1-2-3 (*.wk2)
- Lotus Symphony (*.wr1)
- QuattroPro (*.wq1)
- XML (*.xml)
- MS Access Database (*.mdb)
- JSON JavaScript Object Notation (*.json)
- OpenOffice Spreadsheet (*.ods)

- Export Formats include:

- Paradox (*.db)
- dBase (*.dbf)
- Comma Separated Values (*.csv)
- HTML (*.HTML, *.HTML)
- MS Excel Spreadsheet (*.xls)
- MS Excel Workbook (*.xls)
- MS Word (*.doc)
- Tab Delimited (*.tab)
- SYLK Symbolic Link (*.slk)
- DIF Data Interchange Format (*.dif)
- Lotus 1-2-3 (*.wk1)

QuattroPro (*.wq1)
 SQL Script (*.sql)
 o MSSQL
 o MySQL
 o InerBase/FireBird
 o Oracle
 o PostgreSQL
 o DB2
 o Informix
 o MS Access
 o SQLite
 XML (*.xml)
 MS Windows Clipboard
 Rich Text Format (*.rtf)
 SPSS Format (*.sav)
 Adobe Acrobat Document (*.pdf)
 LDAP Data Interchange Format (*.ldif)
 JSON JavaScript Object Notation (*.json)
 OpenOffice Text Document (*.odt)

1.9.1.1 EXPORT

Specifies that you want to export an R:BASE table or view to an external file.

```
GATEWAY EXPORT type filename [ SELECT clause ] [ OPTION parameter ]
```

Options

filename

Specifies the name of the file to export to.

OPTION

Specifies an additional parameter for the output. Options listed may vary based upon selected format.

You must Use the pipe "|" symbol to separate OPTION parameters if more than one parameter is used.

- ACTION
- ACTIVE_INDEX
- ADD_UTF8_BOM
- BLANK_IF_NULL
- BLANK_IF_ZERO
- COL_NAMES
- DATASET_FIELD_ORDER
- DETECT_FORMULA
- EXTRA_SHEETS
- FOOTER_TEXT
- FORMAT
- HEADER_TEXT
- HTML_PATERN
- LAYOUT
- MESSAGES
- NAMED_NUM
- NAMED_TEXT
- NO_UNICODE_CONVERSION
- PASSWORD
- PASSWORD_TO_OPEN
- QUALIFIER
- RAW_GEN
- REC_SEP
- ROWS_PER_FILE
- SEPARATOR

- **SHEET_INDEX**
- **SHEET_NAME**
- **SHOW_PROGRESS**
- **SPECIFICATION_FILE_NAME**
- **STRUCTURE**
- **STYLE**
- **TAB_NAME**
- **TABLE_NAME**
- **TITLE**
- **WRAP_TEXT**

SELECT clause

Specifies and limits rows of data using the [SELECT](#) command.

type

Specifies one of the following codes for accepted file formats:

<u>Code</u>	<u>Details</u>
DB	Paradox
DBF	DBASE
CSV	ASCII File, Delimiter Comma, Qualifier Double Quote
FIX	ASCII File, Fixed Width
HTM	HTML
XLS	MS Excel Spreadsheet
XLSW	MS Excel Workbook
DOC	MS Word
TAB	ASCII File, Delimiter Tab, Qualifier None
SLK	SYLK Symbolic Link
DIF	DIF Data Interchange Format
WK1	Lotus 1_2_3
WQ1	QuattroPro
SQL	SQL Script
XML	XML
CLBR	MS Windows Clipboard
RTF	Rich Text Format
SAV	SPSS Format
PDF	Adobe Acrobat Portable Document Format
LDIF	LDAP Data Interchange Format
JSON	JavaScript Object Notation
ODT	OpenOffice Text Document

1.9.1.1.1 DB (Paradox)

Syntax:

```
GATEWAY EXPORT DB filename.db +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|MERGE_DATA value +
|SPECIFICATION_FILE_NAME filename +
```

```
|DATASET_FIELD_ORDER value +  
|NO_UNICODE_CONVERSION value +  
|ADD_UTF8_BOM value
```

filename.db

This parameter defines the file name of the target file with exported data using the Paradox format. The file extension must be **db**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT DB D:\DataStore\Employee.DB +
```

```
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT DB D:\DataStore\Contacts.DB +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

Notes:

- This option allows the data transfer using Borland Database Engine (BDE). To use this option, the BDE 32-bit version must be installed on the computer.
- Only ANSI file names are supported for Paradox exports.

1.9.1.1.2 DBF (dBASE)

Syntax:

```
GATEWAY EXPORT DBF filename.dbf +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|MERGE_DATA value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.dbf

This parameter defines the file name of the target file with exported data using the dBASE format. The file extension must be **dbf**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view..

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT DBF D:\DataStore\Employee.DBF +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT DBF D:\DataStore\Contacts.DBF +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.3 CSV (Comma Separated Values)

Syntax:

```
GATEWAY EXPORT CSV filename.csv +
SELECT clause +
OPTION COL_NAMES value +
|QUALIFIER symbol +
|SEPARATOR symbol +
|REC_SEP value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|MERGE_DATA value +
|ESCAPE_QUOTES value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.csv

This parameter defines the file name of the target file with exported data using the Comma Separated Values format. The file extension must be **csv**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

ESCAPE_QUOTES

Specifies if R:BASE exports data and adds an additional double quote character (") if data is encountered where a double quotes exists. The two double quotes allow for greater data compatibility with other program input processes. The default is set to ON. If ESCAPE_QUOTES is set to OFF, double quote characters are exported as is. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

QUALIFIER

Changing the value of this option you can define any qualifier for fields in the target Comma Separated Values text file. The popular value is Quotes (") but you can define any custom character you wish. Use the keyword **PIPE** as the value if you wish to use the pipe character as the qualifier. Use the keyword **NONE** if you wish to specify no text qualifier.

REC_SEP

Defines the separator between lines in the target text file. The popular value is CRLF (#13#10) or CR(#13) but you can define any other custom string. Available Parameters: CR / LF / CRLF

SEPARATOR

Defines the separator of fields in target text file. The popular values are Semicolon (;), Comma (,), Space (#32) but you can define any other character if you wish. If the pipe character is needed to be specified as the separator, then use the keyword **PIPE** as the value.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT CSV D:\TEMP\Employee.CSV +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|QUALIFIER " |SEPARATOR , |REC_SEP CRLF +
|SHOW_PROGRESS ON |ACTION OPENVIEW
```

```
GATEWAY EXPORT CSV G:\Durgin.CSV +
SELECT * FROM QuoteTable +
OPTION COL_NAME ON|SEPARATOR PIPE
```

```
GATEWAY EXPORT CSV D:\DataStore\Contacts.CSV +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.4 FIX (Fixed Width)

Syntax:

```
GATEWAY EXPORT FIX filename.txt +
SELECT clause +
OPTION COL_NAMES value +
|COL_WIDTHS value +
|SPACE_BETWEEN_COLUMNS value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|MERGE_DATA value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.txt

This parameter defines the file name of the target file with exported data using the fixed width format.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

COL_WIDTHS

Indicates column length specifications for values greater than 10 characters. Without the length specification, the default is 10. The parameter accepts a comma delimited list of integers to specify column widths.

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPACE_BETWEEN_COLUMNS

If ON, a space will be added between columns. The default is set to OFF. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT FIX D:\DataStore\Employee.TXT +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|COL_WIDTHS 10,10,10,16,30,30,20,2,10,12,10,12,12,2,60,60 +
|SHOW_PROGRESS ON |ACTION OPENVIEW
```

```
GATEWAY EXPORT FIX D:\DataStore\Contacts.TXT +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.5 HTM(HTML)

Syntax:

```
GATEWAY EXPORT HTM filename.htm +
SELECT clause +
OPTION HTML_PATERN filespec +
|LAYOUT value +
|STYLE value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.htm

This parameter defines the file name of the target file with exported data using the HTML format. The file extension must be either **htm** or **html**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

HTML_PATERN

With this option you can define your own pattern of the HTML file to customize any settings, fonts, colors, metatags, scripts, etc. Using this option you can integrate the generated html_file into a web site. By default, the HTML pattern has a simple structure:

```
<html>
<head>
<meta http_equiv="Content_Type"
content="text/html; charset=windows_1251">
<meta name="GENERATOR" content="RBG Gateway">
<title>Data Exported by R:BASE File Gateway</title>
</head>
<body bgcolor="#FFFFFF">
</body>
```

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

STYLE

This option adds background color to exported data.

Available Parameters:

```
NORMAL
PRICELIST
MSMONEY
BRICK
DESERT
EGGPLANT
LILAC
MAPLE
MARINE
ROSE
SPRUCE
WHEAT
ZEBRA
```

Examples:

```
GATEWAY EXPORT HTM D:\DataStore\Employee.HTML +
SELECT (EmpLName+', '&EmpFName) FROM Employee WHERE EmpStatus = 'Active' +
```

```
OPTION MESSAGES ON|ACTION OPENVIEW
```

```
GATEWAY EXPORT HTM D:\DataStore\Contacts.HTM +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF|STYLE SPRUCE
```

Notes:

The generated data table will be inserted before the </body> tag.

1.9.1.1.6 XLS (MS Excel Spreadsheet)

Syntax:

```
GATEWAY EXPORT XLS filename.xls +
SELECT clause +
OPTION COL_NAMES value +
|TAB_NAME value +
|NAMED_NUM value1 value2 +
|NAMED_TEXT value1 value2 +
|HEADER_TEXT value +
|FOOTER_TEXT value +
|LAYOUT value +
|STYLE value +
|DETECT_FORMULA value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|PASSWORD value +
|PASSWORD_TO_OPEN value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|NAMED_NUM value +
|NAMED_TEXT value +
|WRAP_TEXT value +
|MERGE_DATA value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.xls

This parameter defines the file name of the target file with exported data using the MS Excel Spreadsheet format. The file extension must be **xls**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

DETECT_FORMULA

Specifies that when equal characters exist in table values, which are not actual formulas, formula detection is disabled. The default is set to ON. Available Parameters: ON / OFF

FOOTER_TEXT

The parameter specifies text that will appear as a footer at the bottom of the exported data. Multiple footers can be specified by repeating the parameter. An approximate 250 character limit exists for a spreadsheet footer. If the specified footer exceeds the limit, R:BASE will determine the best length that Excel can accept and truncate the data.

HEADER_TEXT

The parameter specifies text that will appear as a header on top of the exported data. Multiple headers can be specified by repeating the parameter. An approximate 250 character limit exists for a spreadsheet header. If the specified header exceeds the limit, R:BASE will determine the best length that Excel can accept and truncate the data.

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NAMED_NUM

Specifies to pass a number formula to the Excel Names Manager. A formula name and value is passed with the parameter. The parameter can be repeated if multiple number are to be added.

NAMED_TEXT

Specifies to pass a text formula to the Excel Names Manager. A formula name and value is passed with the parameter. The parameter can be repeated if multiple number are to be added.

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

PASSWORD

Specifies the password for the Excel file. Passwords are case sensitive.

PASSWORD_TO_OPEN

Specifies the password for an existing password protected spreadsheet file, to allow appending to the file with the MERGE_DATA option. Passwords are case sensitive.

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

STYLE

This option adds background color to exported data.

Available Parameters:

NORMAL
 PRICELIST
 MSMONEY
 BRICK
 DESERT
 EGGPLANT
 LILAC
 MAPLE
 MARINE
 ROSE
 SPRUCE
 WHEAT
 ZEBRA

TAB_NAME

If defined, will be used as the label for the first tab on MS Excel Spreadsheet. The specified name cannot exceed 31 characters, have the same name as another tab, set as a blank value, or contain special characters (backward slash, forward slash, question mark, asterisk, left bracket, right bracket, colon).

WRAP_TEXT

Specifies to wrap long text values within the output of the spreadsheet.

Examples:

```
GATEWAY EXPORT XLS D:\DataStore\Employee.XLS +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|HEADER_TEXT 'Employee Information' +
|HEADER_TEXT 'September' +
|FOOTER_TEXT ' ' +
|FOOTER_TEXT 'Prepared by Jane Sullivan' +
|SHOW_PROGRESS ON +
|ACTION OPENVIEW
```

```
GATEWAY EXPORT XLS Customers_1203.XLS +
SELECT +
Company AS `Company Name`, +
CustID AS `Customer ID`, +
CustAddress AS `Company Address`, +
```

```
(CustCity+', '&CustState&CustZip) AS `City, State, and Zip Code` +
FROM Customer ORDER by Company +
OPTION COL_NAMES ON +
|TAB_NAME Customer List
```

```
GATEWAY EXPORT XLS D:\DataStore\Contacts.XLS +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF|STYLE SPRUCE
```

```
GATEWAY EXPORT XLS DataCalculation.XLS +
SELECT * FROM DiscountRates +
OPTION COL_NAMES ON +
|TAB_NAME SalesDiscounts +
|NAMED_NUM MinDiscount 0.01 +
|NAMED_NUM MaxDiscount 0.05 +
|NAMED_TEXT DiscountSuffix Disc
```

```
GATEWAY EXPORT XLS BonusRate2018.xls +
SELECT * FROM BonusRate OPTION PASSWORD ApplePie
```

```
GATEWAY EXPORT XLS StaffList.xls +
SELECT DepartmentID,HireDate,LastName FROM Staff +
ORDER BY DepartmentID ASC, LastName ASC +
OPTION TAB_NAME Staff +
|SPECIFICATION_FILE_NAME StaffExp.rgw +
|DATASET_FIELD_ORDER ON
```

Note:

Use this option to export data to a MS Excel spreadsheet. The process does not use the MS Excel application as OLE server unless the password option is specified. If the process uses a password, MS Excel is required.

1.9.1.1.7 XLSW (MS Excel Workbook)

Syntax:

```
GATEWAY EXPORT XLSW filename.xls +
SELECT clause +
OPTION COL_NAMES value +
|TAB_NAME value +
|NAMED_NUM value1 value2 +
|NAMED_TEXT value1 value2 +
|HEADER_TEXT value +
|FOOTER_TEXT value +
|SHEET_INDEX value +
|SHEET_NAME value +
|EXTRA_SHEETS value +
|ACTIVE_INDEX value +
|RAW_GEN value +
|STYLE value +
|DETECT_FORMULA value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|OVERWRITE_SHEET value +
|PASSWORD value +
|PASSWORD_TO_OPEN value +
```

```
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|NAMED_NUM value +
|NAMED_TEXT value +
|WRAP_TEXT value +
|MERGE_DATA value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.xls

This parameter defines the file name of the target file with exported data using the MS Excel Workbook format. The file extension must be **xls** or **xlsx**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ACTIVE_INDEX

Defines the active sheet when the worksheet is opened in Excel. The value is an integer-based index (e.g. 1, 2, etc.).

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

DETECT_FORMULA

Specifies that when equal characters exist in table values, which are not actual formulas, formula detection is disabled. The default is set to ON. Available Parameters: ON / OFF

EXTRA_SHEETS

Specifies whether blank sheets (Sheet2, Sheet3) are automatically included in the exported workbook. Available Parameters: ON / OFF

FOOTER_TEXT

The parameter specifies text that will appear as a footer at the bottom of the exported data. Multiple footers can be specified by repeating the parameter. An approximate 250 character limit exists for a spreadsheet footer. If the specified footer exceeds the limit, R:BASE will determine the best length that Excel can accept and truncate the data.

HEADER_TEXT

The parameter specifies text that will appear as a header on top of the exported data. Multiple headers can be specified by repeating the parameter. An approximate 250 character limit exists for a spreadsheet header. If the specified header exceeds the limit, R:BASE will determine the best length that Excel can accept and truncate the data.

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NAMED_NUM

Specifies to pass a number formula to the Excel Names Manager. A formula name and value is passed with the parameter. The parameter can be repeated if multiple number are to be added.

NAMED_TEXT

Specifies to pass a text formula to the Excel Names Manager. A formula name and value is passed with the parameter. The parameter can be repeated if multiple number are to be added.

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

OVERWRITE_SHEET

Specifies an existing sheet with the same name will be overwritten with the new dataset. The default is set to OFF. Available Parameters: ON / OFF

PASSWORD

Specifies the password for the Excel file. MS Excel is required in order to use the PASSWORD parameter. Passwords are case sensitive.

PASSWORD_TO_OPEN

Specifies the password for an existing password protected spreadsheet file, to allow appending to the file with the MERGE_DATA option. Passwords are case sensitive.

RAW_GEN

Specifies the program to generate the workbook. When ON, R:BASE will generate the workbook, which performs the export process much faster. When OFF, Excel will generate the workbook as native MS output. When R:BASE creates the output, only the most important content is placed in the file, where the workbook is intended to mimic Excel output. The R:BASE output can be considered non-native, yet less bloated. If the PASSWORD parameter is specified, Excel is always used to generate the workbook, regardless of the RAW_GEN value. The default is set to OFF. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHEET_INDEX

Specifies the workbook sheet/tab to place the exported data. The index value begins with 1. It is recommended to use 1 based incrementing values for multiple exports to separate sheets of a workbook. If the desired output is to display the sheet captions as: [May][June][July], the code should be written as:

```
GATEWAY ... SHEET_NAME May|SHEET_INDEX 1
GATEWAY ... SHEET_NAME June|SHEET_INDEX 2
GATEWAY ... SHEET_NAME July|SHEET_INDEX 3
```

SHEET_NAME

Assigns a custom text label to the workbook sheet. The specified name cannot exceed 31 characters, have the same name as another sheet, set as a blank value, or contain special characters (backward slash, forward slash, question mark, asterisk, left bracket, right bracket, colon).

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

STYLE

This option adds background color to exported data.

Available Parameters:

NORMAL
 PRICELIST
 MSMONEY
 BRICK
 DESERT
 EGGPLANT
 LILAC
 MAPLE
 MARINE
 ROSE
 SPRUCE
 WHEAT
 ZEBRA

TAB_NAME

If defined, will be used as the label for the first tab on MS Excel Spreadsheet. The specified name cannot exceed 31 characters, have the same name as another tab, set as a blank value, or contain special characters (backward slash, forward slash, question mark, asterisk, left bracket, right bracket, colon).

WRAP_TEXT

Specifies to wrap long text values within the output of the spreadsheet.

Examples:

```
GATEWAY EXPORT XLSW D:\DataStore\Employee.XLS +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|TAB_NAME EMPLOYEES +
|HEADER_TEXT 'Employee Information' +
|HEADER_TEXT 'September' +
|FOOTER_TEXT ' ' +
|FOOTER_TEXT 'Prepared by Jane Sullivan' +
|STYLE ZEBRA +
|SHEET_NAME Employee Info +
|SHEET_INDEX 2 +
|ACTIVE_INDEX 1 +
|SHOW_PROGRESS ON +
|MERGE_DATA ON +
|RAW_GEN ON +
|ACTION OPENVIEW
```

```
GATEWAY EXPORT XLSW Customers_1203.XLS +
SELECT +
Company AS `Company Name`, +
CustID AS `Customer ID`, +
CustAddress AS `Company Address`, +
(CustCity+', '&CustState&CustZip) AS `City, State, and Zip Code` +
FROM Customer ORDER by Company +
OPTION COL_NAMES ON +
|TAB_NAME Customer List
```

```
GATEWAY EXPORT XLSW D:\DataStore\Contacts.XLSX +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

```
GATEWAY EXPORT XLSW DataCalculation.XLSX +
SELECT * FROM DiscountRates +
OPTION COL_NAMES ON +
|TAB_NAME SalesDiscounts +
|NAMED_NUM MinDiscount 0.01 +
|NAMED_NUM MaxDiscount 0.05 +
|NAMED_TEXT DiscountSuffix Disc
```

```
GATEWAY EXPORT XLSW BonusRate2018.xlsx +
SELECT * FROM BonusRate OPTION PASSWORD ApplePie
```

Notes:

- Use this option to export data to a MS Excel workbook. The process uses the MS Excel application as OLE server unless the RAW_GEN parameter is set to ON, or the PASSWORD parameter is specified. If the process uses a password, MS Excel is required.
- When providing a file name to export to, you will be required to specify the complete path.

1.9.1.1.8 DOC (MS Word Document)

Syntax:

```
GATEWAY EXPORT DOC filename.doc +
SELECT clause +
OPTION TITLE value +
|COL_NAMES value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.doc

This parameter defines the file name of the target file with exported data using the MS Word document format. The file extension must be **doc**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

TITLE

If defined, will be the page header on each page of the document.

Examples:

```
GATEWAY EXPORT DOC D:\TEMP\Employee.DOC +  
SELECT * FROM Employee WHERE EmpStatus = 'Active' +  
OPTION TITLE 'Employee List' |COL_NAMES ON +  
|SHOW_PROGRESS ON |ACTION OPENVIEW
```

```
GATEWAY EXPORT DOC G:\Durgin.DOC +  
SELECT * FROM QuoteTable +
```

```
OPTION COL_NAME ON
```

```
GATEWAY EXPORT DOC D:\DataStore\Contacts.DOC +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

Notes:

- This option uses the MS Word application OLE server to export data and create the DOC file. To use this option, you need to have MS Word installed on your computer. The time lapse for the export process depends on the record sets.
- As a faster method, it is suggested to use the [RTF](#) export option.

1.9.1.1.9 TAB (Tab Separated Values)

Syntax:

```
GATEWAY EXPORT TAB filename.tab +
SELECT clause +
OPTION COL_NAMES value +
|QUALIFIER value +
|SEPARATOR value +
|REC_SEP value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|MERGE_DATA value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.tab

This parameter defines the file name of the target file with exported data using the Tab Separated Values format. The file extension must be **tab**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

ESCAPE_QUOTES

Specifies if R:BASE exports data and adds an additional double quote character (") if data is encountered where a double quotes exists. The two double quotes allow for greater data compatibility with other program input processes. The default is set to ON. If ESCAPE_QUOTES is set to OFF, double quote characters are exported as is. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

MERGE_DATA

If ON, the exported data will be added to the specified file, if exists. The default is set to OFF. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

QUALIFIER

Changing the value of this option you can define any qualifier for fields in the target Comma Separated Values text file. The popular value is Quotes (") but you can define any custom character you wish. Use the keyword **PIPE** as the value if you wish to use the pipe character as the qualifier. Use the keyword **NONE** if you wish to specify no text qualifier.

REC_SEP

Defines the separator between lines in the target text file. The popular value is CRLF (#13#10) or CR(#13) but you can define any other custom string. Available Parameters: CR / LF / CRLF

SEPARATOR

Defines the separator of fields in target text file. The popular values are Semicolon (;), Comma (,), Space (#32) but you can define any other character if you wish. If the pipe character is needed to be specified as the separator, then use the keyword **PIPE** as the value.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Example:

```
GATEWAY EXPORT TAB D:\TEMP\Employee.TAB +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|QUALIFIER " |SEPARATOR , |REC_SEP CRLF +
|SHOW_PROGRESS ON |ACTION OPENVIEW
```

```
GATEWAY EXPORT TAB G:\Durgin.TAB +
SELECT * FROM QuoteTable +
OPTION COL_NAME ON|SEPARATOR PIPE
```

```
GATEWAY EXPORT TAB D:\DataStore\Contacts.TAB +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.10 SLK (Symbolic Link)

Syntax:

```
GATEWAY EXPORT SLK filename.slk +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.slk

This parameter defines the file name of the target file with exported data using the Symbolic Link format. The file extension must be **slk**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT SLK D:\DataStore\Employee.SLK +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT SLK D:\DataStore\Contacts.SLK +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.11 DIF (Data Interchange Format)

Syntax:

```
GATEWAY EXPORT DIF filename.dif +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.dif

This parameter defines the file name of the target file with exported data using the Data Interchange Format. The file extension must be **dif**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT DIF D:\DataStore\Employee.DIF +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT DIF D:\DataStore\Contacts.DIF +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.12 WK1 (Lotus)

Syntax:

```
GATEWAY EXPORT WK1 filename.wk1 +
```

```

SELECT clause +
OPTION COL_NAMES value +
| LAYOUT value +
| SHOW_PROGRESS value +
| MESSAGES value +
| ACTION value +
| BLANK_IF_NULL value +
| BLANK_IF_ZERO value +
| ROWS_PER_FILE value +
| SPECIFICATION_FILE_NAME filename +
| DATASET_FIELD_ORDER value +
| NO_UNICODE_CONVERSION value +
| ADD_UTF8_BOM value

```

filename.wk1

This parameter defines the file name of the target file with exported data using the Lotus 1-2-3 format. The file extension must be **wk1**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT WK1 D:\DataStore\Employee.WK1 +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT WK1 D:\DataStore\Contacts.WK1 +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.13 WQ1 (QuattroPro)

Syntax:

```
GATEWAY EXPORT WQ1 filename.wq1 +
SELECT clause +
OPTION COL_NAMES value +
|LAYOUT value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.wq1

This parameter defines the file name of the target file with exported data using the Quattro Pro format. The file extension must be **wq1**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT WQ1 D:\DataStore\Employee.WQ1 +  
SELECT * FROM Employee WHERE EmpStatus = 'Active' +  
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT WQ1 D:\DataStore\Contacts.WQ1 +  
SELECT * FROM Contacts +  
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.14 SQL (SQL Script)

Syntax:

```

GATEWAY EXPORT SQL filename.sql +
SELECT clause +
OPTION QUALIFIER symbol +
|FORMAT value +
|STRUCTURE value +
|TABLE_NAME value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value

```

filename.sql

This parameter defines the file name of the target file with exported data using a SQL Script format. The file extension must be **sql**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

FORMAT

Specifies the database format type. Below lists valid values for the FORMAT parameter. If a FORMAT is not specified or an invalid value is used then a generic SQL format will be used.

Format Type	Parameter Value
DB2	DB2
InterBase, Firebird	INTERBASE, IB, FIREBIRD, FB
Access	MSACCESS, ACCESS
SQL Server	MSSQL, SQL, SQLSERVER
MySQL	MYSQL
Oracle	ORACLE
PostgreSQL	POSTGRES, PG

SQLite

SQLITE

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

QUALIFIER

Changing the value of this option you can define any qualifier for fields in the target Comma Separated Values text file. The popular value is Quotes (") but you can define any custom character you wish. Use the keyword **PIPE** as the value if you wish to use the pipe character as the qualifier. Use the keyword **NONE** if you wish to specify no text qualifier.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

STRUCTURE

Adds a CREATE TABLE command in the script. Available Parameters: ON / OFF

TABLE_NAME

Sets a table name different from the table name in the SELECT command. As the source is a query and supports JOINS, the TABLE_NAME parameter is available to specify a table after the FROM keyword is used.

Examples:

```
GATEWAY EXPORT SQL D:\TEMP\Employee.SQL +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT SQL D:\DataStore\Contacts.SQL +
SELECT * FROM Contacts +
OPTION FORMAT ORACLE|STRUCTURE ON|TABLE_NAME ContactTable
|BLANK_IF_ZERO ON|MESSAGES OFF
```

Notes:

Each record will include INSERT INTO statements which can be executed in the supported SQL compliant databases.

1.9.1.1.15 XML (Extended Markup Language)

Syntax:

```
GATEWAY EXPORT XML filename.xml +
SELECT clause +
OPTION SHOW_PROGRESS value +
```

```
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.xml

This parameter defines the file name of the target file with exported data using the XML format. The file extension must be **xml**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT XML D:\DataStore\Employee.XML +  
SELECT * FROM Employee WHERE EmpStatus = 'Active' +  
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT XML D:\DataStore\Contacts.XML +  
SELECT * FROM Contacts +  
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.16 CLBR (MS Windows Clipboard)

Syntax:

```
GATEWAY EXPORT CLBR +  
SELECT clause +  
OPTION SHOW_PROGRESS value +  
|MESSAGES value +  
|ACTION value +  
|BLANK_IF_NULL value +  
|BLANK_IF_ZERO value +  
|SPECIFICATION_FILE_NAME filename +  
|DATASET_FIELD_ORDER value +  
|NO_UNICODE_CONVERSION value +  
|ADD_UTF8_BOM value
```

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT CLBR +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON +
|ACTION OPENVIEW
```

```
GATEWAY EXPORT CLBR +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

Notes:

- Use this option to generate exported data as MS Windows Clipboard Data.
- ACTION OPENVIEW option will launch MS Clipbook Viewer. Then, use Paste to view the data.
- Refer to the MS Clipbook Viewer for more details and handling exported data.

1.9.1.1.17 RTF (Rich Text Format)

Syntax:

```
GATEWAY EXPORT RTF filename.rtf +
SELECT clause +
OPTION TITLE value +
|COL_NAMES value +
|LAYOUT value +
|STYLE value +
|SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.rtf

This parameter defines the file name of the target file with exported data using the Rich Text Format. The file extension must be **rtf**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

STYLE

This option adds background color to exported data.

Available Parameters:

NORMAL
PRICELIST
MSMONEY
BRICK
DESERT
EGGPLANT
LILAC

MAPLE
MARINE
ROSE
SPRUCE
WHEAT
ZEBRA

TITLE

If defined, will be the page header on each page of the document.

Examples:

```
GATEWAY EXPORT RTF D:\TEMP\Employee.RTF +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION TITLE 'Employee List' |COL_NAMES ON +
|SHOW_PROGRESS ON |ACTION OPENVIEW
```

```
GATEWAY EXPORT RTF G:\Durgin.RTF +
SELECT * FROM QuoteTable +
OPTION COL_NAME ON
```

```
GATEWAY EXPORT RTF D:\DataStore\Contacts.RTF +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF|STYLE SPRUCE
```

1.9.1.1.18 SAV (SPSS Format)

Syntax:

```
GATEWAY EXPORT SAV filename.sav +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.sav

This parameter defines the file name of the target file with exported data using the SPSS format. The file extension must be **sav**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT SAV D:\DataStore\Employee.SAV +  
SELECT * FROM Employee WHERE EmpStatus = 'Active' +  
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT SAV D:\DataStore\Contacts.SAV +  
SELECT * FROM Contacts +  
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.19 PDF (Adobe Acrobat Portable Document)**Syntax:**

```
GATEWAY EXPORT PDF filename.pdf +  
SELECT clause +  
OPTION COL_NAMES value +  
|LAYOUT value +  
|SHOW_PROGRESS value +  
|MESSAGES value +  
|ACTION value +  
|BLANK_IF_NULL value +  
|BLANK_IF_ZERO value +  
|SPECIFICATION_FILE_NAME filename +  
|DATASET_FIELD_ORDER value +  
|NO_UNICODE_CONVERSION value +  
|ADD_UTF8_BOM value
```

filename.pdf

This parameter defines the file name of the target file with exported data using the Adobe Acrobat Portable Document Format. The file extension must be **pdf**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:


```
GATEWAY EXPORT PDF D:\DataStore\Employee.PDF +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON|SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT PDF D:\DataStore\Contacts.PDF +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF|ACTION OPENVIEW
```

1.9.1.1.20 LDIF (Data Interchange Format)

Syntax:

```
GATEWAY EXPORT LDIF filename.dif +
SELECT clause +
OPTION SHOW_PROGRESS value +
|BLANK_IF_ZERO value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.dif

This parameter defines the file name of the target file with exported data using the LDAP Data Interchange Format. The file extension must be **dif**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION

ACTION

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT LDIF D:\DataStore\Employee.DIF +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON|ACTION OPENVIEW
```

```
GATEWAY EXPORT LDIF D:\DataStore\Contacts.DIF +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.21 JSON (JavaScript Object Notation)

Syntax:

```
GATEWAY EXPORT JSON filename.json +
SELECT clause +
OPTION SHOW_PROGRESS value +
|MESSAGES value +
|ACTION value +
|BLANK_IF_NULL value +
|BLANK_IF_ZERO value +
|ROWS_PER_FILE value +
|SPECIFICATION_FILE_NAME filename +
|DATASET_FIELD_ORDER value +
|NO_UNICODE_CONVERSION value +
|ADD_UTF8_BOM value
```

filename.json

This parameter defines the file name of the target file with exported data using the JSON format. The file extension must be **json**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

DATASET_FIELD_ORDER

Specifies the export will follow the SELECT clause column order, rather than the field order in the specification file, when a specification file is used as an OPTION parameter. The default is set to OFF. Available Parameters: ON / OFF

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT JSON D:\DataStore\Employee.JSON +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION SHOW_PROGRESS ON|MESSAGES ON
```

```
GATEWAY EXPORT JSON D:\DataStore\Contacts.JSON +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

1.9.1.1.22 ODT (OpenOffice Text Document)

Syntax:

```
GATEWAY EXPORT ODT filename.odt +
```

```

SELECT clause +
OPTION COL_NAMES value +
| LAYOUT value +
| SHOW_PROGRESS value +
| MESSAGES value +
| ACTION value +
| BLANK_IF_NULL value +
| BLANK_IF_ZERO value +
| ROWS_PER_FILE value +
| SPECIFICATION_FILE_NAME filename +
| NO_UNICODE_CONVERSION value +
| ADD_UTF8_BOM value

```

filename.odt

This parameter defines the file name of the target file with exported data using the OpenOffice Text Document format. The file extension must be **odt**.

SELECT clause

Use the [SELECT](#) command to identify rows of data from a table or view.

OPTION**ACTION**

Defines the default action which will be processed after export is complete, for example, to open the target export file in view/edit mode or send this exported file by email as attachment. Available Parameter: OPENVIEW

ADD_UTF8_BOM

Specifies whether to add a UTF8 BOM to an exported UTF8 string. Available Parameters: ON / OFF

BLANK_IF_NULL

If ON, will export NULL values as blank values. If OFF, will export NULL values as the database NULL setting. The default is set to OFF. Available Parameters: ON / OFF

BLANK_IF_ZERO

If ON, will export zero values as blank values. If OFF, will export zero values as zero. The default is set to OFF. Available Parameters: ON / OFF

COL_NAMES

If ON, will add a first row with column names. If OFF, will only export data. Available Parameters: ON / OFF

LAYOUT

Specifies the layout in which the exported data is arranged. The columnar option arranges columns across the top. Reversed columnar arranges columns along the left side. The tabular option arranges data that looks like a table. Available Parameters: COLUMNAR / REVERSED_COLUMNAR / TABULAR

MESSAGES

If ON, during export process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire export process will be in "silent" mode. Available Parameters: ON / OFF

NO_UNICODE_CONVERSION

Exports Unicode data as a UTF8 string to fit in the target field, ensuring no data is lost. If OFF, the Unicode data will be converted to ANSI and characters from source that is higher than 0x127 will become '?'. As not all target formats support Unicode and this field is for cases when source is Unicode and destination is normal text field. Available Parameters: ON / OFF

ROWS_PER_FILE

Specifies if the records are broken up into separate files based on the quantity of records in each file. If this parameter is not zero, multiple files may be generated.

SHOW_PROGRESS

Indicates whether to display status dialog during export process. Set SHOW_PROGRESS to ON if you wish to show an animated status dialog during the export process. Set SHOW_PROGRESS to OFF if you wish to export data without animated status dialog. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for exporting. The export must be performed once using the graphic user interface "Export Wizard" and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY EXPORT ODT D:\TEMP\Employee.odt +
SELECT * FROM Employee WHERE EmpStatus = 'Active' +
OPTION COL_NAMES ON +
|SHOW_PROGRESS ON +
|ACTION OPENVIEW
```

```
GATEWAY EXPORT ODT G:\Durgin.odt +
SELECT * FROM QuoteTable +
OPTION COL_NAME ON
```

```
GATEWAY EXPORT ODT D:\DataStore\Contacts.odt +
SELECT * FROM Contacts +
OPTION BLANK_IF_ZERO ON|MESSAGES OFF
```

Notes:

- This option uses the OpenOffice application OLE server to export data and create the ODT file. To use this option, you need to have OpenOffice installed on your computer.

1.9.1.2 Export Dialog

The File Gateway utility can be launched with the available export options preselected, allowing for easier file transfers for end users. The **OPTION** keyword must immediately follow the EXPORT keyword in order for the default options to be assigned.

GATEWAY EXPORT OPTION <default_parameter> <value>

Default Parameter	Value
DEFAULT_TABLE	value
ONLY_TABLE	value
DEFAULT_FORMAT	WIZARD, DB, DBF, CSV, HTM, HTML, XLS, XLSX, XLSW, DOC, SLK, DIF, WK1, WQ1, SQL, XML, CLIP, CLIPBOARD, RTF, SAV, PDF, LDIF, JSON, ODT
DEFAULT_SQL_FORMAT	GENERIC, DB2, INTERBASE, IB, FIREBIRD, FB, MSACCESS, ACCESS, MSSQL, SQL, SQLSERVER, MYSQL, ORACLE, POSTGRESQL, POSTGRES, PG, SQLITE
DEFAULT_INCLUDE_STRUCTURE	ON/OFF
DEFAULT_FILENAME	value
DEFAULT_WHERE	value
DEFAULT_SHOW_PROGRESS	ON/OFF
DEFAULT_ZERO_AS_BLANK	ON/OFF
DEFAULT_NULL_VALUE_AS_BLANK	ON/OFF

DEFAULT_APPEND	ON/OFF
DEFAULT_PASSWORD	value
DEFAULT_PASSWORD_TO_OPEN	value
DEFAULT_LAUNCH_AFTER_EXPORT	ON/OFF
DEFAULT_NO_UNICODE_TO_ANSI	ON/OFF
DEFAULT_ADD_UTF8_BOM	ON/OFF

Example 01:

-- launches the Gateway Utility with only the Export tab displayed
 GATEWAY EXPORT

Example 02:

-- launches the Gateway Utility Export tab with Customer preselected as the default table, and prevents the selection of other tables

```
GATEWAY EXPORT OPTION DEFAULT_TABLE Customer +
|ONLY_TABLE Customer
```

Example 03:

-- launches the Gateway Utility Export tab with History preselected as the default table, DBF as the file format, a check in the append check box, and with the file name specified

```
GATEWAY EXPORT OPTION DEFAULT_TABLE History +
|DEFAULT_FORMAT DBF +
|DEFAULT_APPEND ON +
|DEFAULT_FILENAME History.dbf
```

Example 04:

-- launches the Gateway Utility Export tab with Transactions as the preselected table, SQL as the file format, and SQL Server as the SQL format

```
GATEWAY EXPORT OPTION DEFAULT_TABLE Transactions +
|DEFAULT_FORMAT SQL +
|DEFAULT_SQL_FORMAT MSSQL
```

Example 05:

-- launches the Gateway Utility Export tab with Transactions as the preselected table, SQL as the file format, SQL Server as the SQL format including the table structure, and with the file name specified

```
GATEWAY EXPORT OPTION DEFAULT_TABLE Transactions +
|DEFAULT_FORMAT SQL +
|DEFAULT_SQL_FORMAT MSSQL +
|DEFAULT_INCLUDE_STRUCTURE ON +
|DEFAULT_FILENAME Transactions.sql
```

Example 06:

-- launches the Gateway Utility Export tab with Customer as the preselected table, XLS as the file format, a WHERE Clause specified, the progress display enabled, and the option enabled to launch file after the export completes

```
GATEWAY EXPORT OPTION DEFAULT_TABLE Customer +
|DEFAULT_FORMAT XLS +
|DEFAULT_WHERE WHERE CustState IN (CA,PA) ORDER BY Company +
|DEFAULT_FILENAME Customer.xls +
|DEFAULT_SHOW_PROGRESS ON +
|DEFAULT_LAUNCH_AFTER_EXPORT ON
```

Example 07:

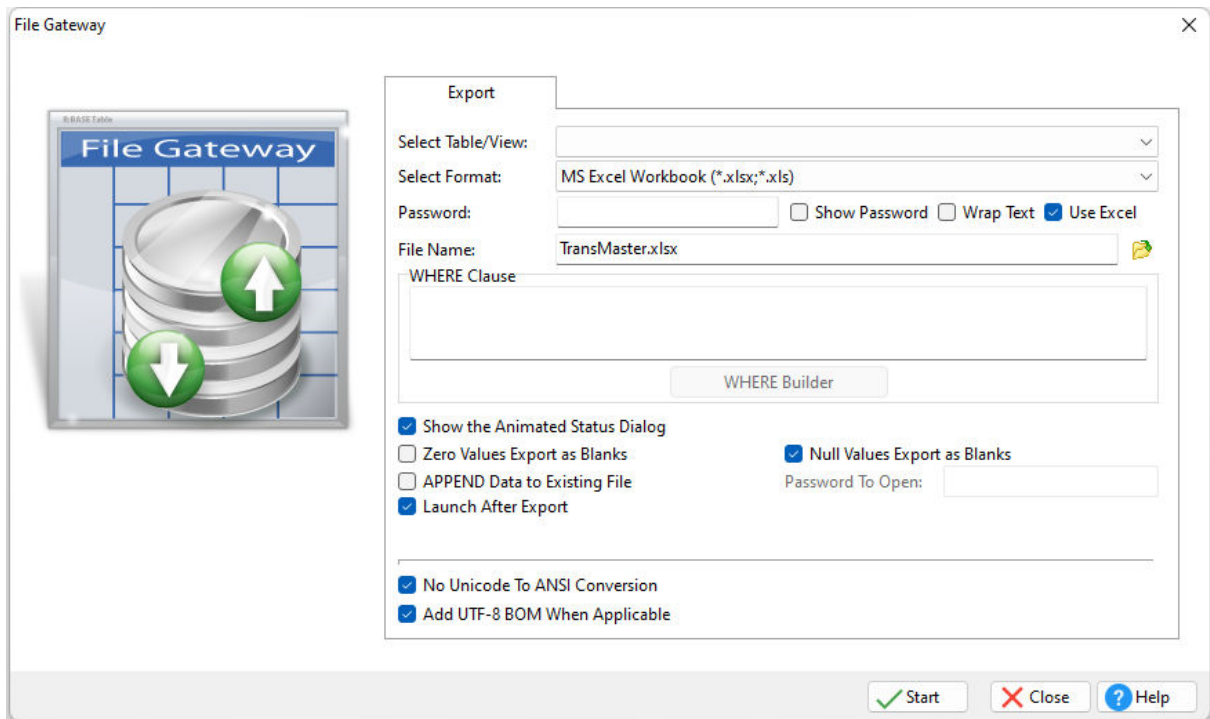
-- launches the Gateway Utility Export tab with TransMaster as the preselected table, XLSX as the file format, the file name specified, NULLs set to display as blank, exports Unicode data as a UTF-8 string, and specifies to add a UTF-8 BOM to exported UTF-8 strings

```
GATEWAY EXPORT OPTION DEFAULT_TABLE TransMaster +
|DEFAULT_FORMAT XLSX +
|DEFAULT_FILENAME TransMaster.xlsx +
```

```

| DEFAULT_NULL_VALUE_AS_BLANK ON +
| DEFAULT_NO_UNICODE_TO_ANSI ON +
| DEFAULT_ADD_UTF8_BOM ON

```



1.9.1.3 IMPORT

Specifies that you want to import an external file to an R:BASE table.

```

GATEWAY IMPORT type filename { APPEND
                                APPEND_UPDATE
                                CREATE
                                REPLACE } tblview [ OPTION parameter ]
                                [ TEMPORARY ]

```

Options

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter

CREATE

Creates a destination table with the same name as the imported file.

filename

Specifies the name of the file to import.

OPTION

Specifies an additional parameter for the input. Options listed may vary based upon selected format. You must Use the pipe "|" symbol to separate OPTION parameters if more than one parameter is used.

- **ADD_MAPPING**
- **AUTONUM**
- **COLUMN_COUNT**
- **ERROR_FILE**
- **ESCAPE_QUOTES**
- **FIRST_ROW**
- **FLATTEN_STRUCTURE**
- **KEYS**
- **LAST_ROW**
- **MESSAGES**
- **NODE_PATH**
- **PASSWORD**
- **PERCENT_MODE**
- **PROCESS_ATTRIBUTES**
- **QUALIFIER**
- **REC_SEP**
- **SEPARATOR**
- **SHEET_INDEX**
- **SHOW_PROGRESS**
- **SOURCE_TABLE**
- **SPECIFICATION_FILE_NAME**

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

type

Specifies one of the following codes for accepted file formats:

<u>Code</u>	<u>Details</u>
<u>DB</u>	Paradox
<u>DBF</u>	dBASE
<u>TXT</u>	ASCII File, Delimiter None, Qualifier None
<u>CSV</u>	ASCII File, Delimiter Comma, Qualifier Double Quote
<u>QSV</u>	ASCII File, Delimiter Comma, Qualifier Single Quote
<u>ISV</u>	ASCII File, Delimiter Semicolon, Qualifier Single Quote
<u>TAB</u>	ASCII File, Delimiter Tab, Qualifier None
<u>TIL</u>	ASCII File, Delimiter Tilde, Qualifier None
<u>CUS</u>	ASCII File, Delimiter and Qualifier based on DB Settings, or custom SEPARATOR and QUALIFIER values
<u>XLS</u>	MS Excel
<u>WKS</u>	Lotus 1-2-3
<u>HTM</u>	HTML
<u>WQ1</u>	QuattroPro
<u>XML</u>	XML Format
<u>MDB</u>	MS Access Database
<u>JSON</u>	JavaScript Object Notation
<u>ODS</u>	OpenOffice Spreadsheet

1.9.1.3.1 DB (Paradox)

Syntax:

```
GATEWAY IMPORT DB +  
filename.db {APPEND/APPEND_UPDATE/CREATE/REPLACE} +  
<tablename> +  
OPTION SPECIFICATION_FILE_NAME filename +  
| COLUMN_COUNT value +  
| KEYS value +  
| ADD_MAPPING value +  
| AUTONUM value +  
| FIRST_ROW n +  
| LAST_ROW n +  
| SHOW_PROGRESS value +  
| MESSAGES value +  
| ERROR_FILE EXCEPT.DAT
```

filename.db

This parameter defines the file name of the target file with data using the Paradox format. File extension must be **db**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row(s) to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT DB D:\DataStore\Contacts.DB CREATE ContactsImport OPTION FIRST_ROW 2
```

```
GATEWAY IMPORT DB D:\DataStore\Transactions.DB APPEND TransLog OPTION SHOW_PROGRESS  
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT DB Customer.DB APPEND Customer +  
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

Notes:

- This option allows the data transfer using Borland Database Engine (BDE). To use this option, the BDE 32-bit version must be installed on the computer.
- Only ANSI file names are supported for Paradox imports.

1.9.1.3.2 DBF (dBASE)

Syntax:

```
GATEWAY IMPORT DBF +  
filename.dbf {APPEND/APPEND_UPDATE/CREATE/REPLACE} +  
<tablename> +  
OPTION SPECIFICATION_FILE_NAME filename +  
|COLUMN_COUNT value +  
|KEYS value +  
|ADD_MAPPING value +  
|AUTONUM value +  
|FIRST_ROW n +  
|LAST_ROW n +  
|SHOW_PROGRESS value +  
|MESSAGES value +  
|ERROR_FILE EXCEPT.DAT
```

filename.dbf

This parameter defines the file name of the target file with data using the dBASE format. File extension must be **dbf**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT DBF D:\DataStore\Contacts.DBF CREATE ContactsImport OPTION FIRST_ROW 2
```

```
GATEWAY IMPORT DBF D:\DataStore\Transactions.DBF APPEND TransLog OPTION SHOW_PROGRESS  
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT DBF Customer.DBF APPEND Customer +  
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

Note:

This option allows the data transfer using Borland Database Engine (BDE). To use this option, the BDE 32-bit version must be installed on the computer.

1.9.1.3.3 ASCII Text (*.txt, *.csv, *.tab)

Syntax:

```
GATEWAY IMPORT TXT +  
filename.txt {APPEND/APPEND_UPDATE/CREATE/REPLACE} +  
<tablename> +  
OPTION SPECIFICATION_FILE_NAME filename +  
|COLUMN_COUNT value +  
|KEYS value +  
|ADD_MAPPING value +  
|AUTONUM value +  
|FIRST_ROW n +  
|LAST_ROW n +  
|QUALIFIER value +  
|SEPARATOR value +  
|REC_SEP value +  
|ESCAPE_QUOTES value +  
|SHOW_PROGRESS value +  
|MESSAGES value +  
|ERROR_FILE EXCEPT.DAT
```

Code	Delimiter	Qualifier
TXT	None	None
CSV	Comma	Double Quotes
QSV	Comma	Single Quotes
ISV	Semicolon	Single Quotes
TAB	Tab	None
TIL	Tilde	None
CUS	Database DELIMIT setting or SEPARATOR parameter	Database QUOTES setting or QUALIFIER parameter

filename.txt

This parameter defines the file name of the target file with data using the ASCII format.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

ESCAPE_QUOTES

Specifies if R:BASE distinguishes between double quote characters (") embedded in data and double quote characters used to delimit the fields during the import process. When a CSV file contains a data segment with a double quote character, R:BASE expects it to be doubled (e. g. the data 3"" gasket will be imported as 3" gasket). If ESCAPE_QUOTES is set to OFF, double quote characters in the imported file are always interpreted as field delimiters. The default is set to ON. Available Parameters: ON / OFF

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

QUALIFIER

Changing the value of this option you can define any text qualifier for fields in the target file. The popular value is Quotes (") but you can define any custom character you wish. Use the keyword **PIPE** as the value if you wish to use the pipe character as the qualifier, or place double pipes. QUALIFIER is only supported for the CUS import format.

REC_SEP

Specifies the record separator in the target file. The common value is CRLF (carriage return + line feed, which moves the parser down to the next line and then to the beginning of the line. A custom record separator can be specified. Available Parameters: CRLF / CR / LF / <Custom>

SEPARATOR

Defines the separator/delimiter of fields in target text file. The popular values are semicolon (;), comma (,), and space (CHAR(32)), but you can define any custom character if you wish. If the pipe character is needed to be specified as the separator, use the **PIPE** keyword as the value, or place

double pipes. SEPARATOR is only supported for the CUS import format. Available Parameters: COMMA / TAB / SEMICOLON / SPACE / PIPE / NONE / <Custom>

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT CSV D:\DataStore\Titles_Import.CSV APPEND_UPDATE Titles OPTION KEYS
EmpTID|FIRST_ROW 2
```

```
GATEWAY IMPORT TAB D:\DataStore\Contacts.TAB CREATE ContactsImport OPTION
SHOW_PROGRESS ON
```

```
GATEWAY IMPORT TXT D:\DataStore\Transactions.TXT APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF|AUTONUM ON
```

```
GATEWAY IMPORT CSV Final22.csv APPEND FinalTR +
OPTION FIRST_ROW 2 +
|ADD_MAPPING ID=Field1 +
|ADD_MAPPING Tax=Field2 +
|ADD_MAPPING Code=Field3 +
|ADD_MAPPING School=Field4 +
|ADD_MAPPING Owner=Field5
```

1.9.1.3.4 XLS (MS Excel)

Syntax:

```
GATEWAY IMPORT XLS +
filename.ext {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION SHEET_INDEX value +
|ADD_MAPPING value +
|SPECIFICATION_FILE_NAME filename +
|COLUMN_COUNT value +
|KEYS value +
|AUTONUM value +
|FIRST_ROW n +
|LAST_ROW n +
|PERCENT_MODE value +
|SHOW_PROGRESS value +
|MESSAGES value +
|PASSWORD value +
|ERROR_FILE EXCEPT.DAT
```

filename.ext

This parameter defines the file name of the target file with data using the MS Excel format. The file extension can be **xls**, **xlsx**, or **xlsm**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

PASSWORD

Specifies the possible password for the Excel import file. Passwords are case sensitive.

PERCENT_MODE

Specifies how percentage cells are treated, as the XLS and XLSX store the values differently. Available Parameters: DEFAULT / 100 / RAW

- DEFAULT - the default behavior is used for importing. XLS stores the value as "9.5" where XLSX stores the value as "0.095".
- 100 - the imported data will be "9.5", where the stored value is multiplied by 100.
- RAW - the imported data will be "0.095". Note that percentage values are stored as (raw) decimals.

SHEET_INDEX value

This parameter is used to specify a specific sheet of the spreadsheet file, starting with 0.

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT XLS D:\DataStore\Contacts.XLS CREATE ContactsImport OPTION SHEET_INDEX
0|FIRST_ROW 2
```

```
GATEWAY IMPORT XLS D:\DataStore\Transactions.XLSX APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT XLS Customer.XLS APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

```
GATEWAY IMPORT XLS +
Jan2020-City-Rates.xlsx REPLACE +
tTaxTableImport +
OPTION SHEET_INDEX 0 +
|AUTONUM OFF +
|FIRST_ROW 7 +
|SHOW_PROGRESS ON +
|MESSAGES ON +
|PERCENT_MODE RAW +
|ERROR_FILE EXCEPT.DAT
```

Notes:

Use this option to import data from a MS Excel file. The process does not use the MS Excel application as OLE server unless the password option is specified. If the process uses a password, MS Excel is required.

1.9.1.3.5 WKS (Lotus 1-2-3)

Syntax:

```
GATEWAY IMPORT WKS +
filename.wk1 {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION SHEET_INDEX 0 +
|SPECIFICATION_FILE_NAME filename +
|COLUMN_COUNT value +
|KEYS value +
|ADD_MAPPING value +
|AUTONUM value +
|FIRST_ROW n +
|LAST_ROW n +
|SHOW_PROGRESS value +
|MESSAGES value +
```


|ERROR_FILE EXCEPT.DAT

filename.wk1

This parameter defines the file name of the target file with data using the Lotus 1-2-3 format. File extension must be **wk1**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHEET_INDEX value

This parameter is used to specify a specific sheet of the spreadsheet file, starting with 0.

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT WKS D:\DataStore\Contacts.WK1 CREATE ContactsImport OPTION SHEET_INDEX
0|FIRST_ROW 2
```

```
GATEWAY IMPORT WKS D:\DataStore\Transactions.WK1 APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT WKS Customer.WKS APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

1.9.1.3.6 HTM (HTML)

Syntax:

```
GATEWAY IMPORT HTM +
filename.htm {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION SPECIFICATION_FILE_NAME filename +
| COLUMN_COUNT value +
| KEYS value +
| ADD_MAPPING value +
| AUTONUM value +
| FIRST_ROW n +
| LAST_ROW n +
| SHOW_PROGRESS value +
| MESSAGES value +
| ERROR_FILE EXCEPT.DAT
```

filename.htm

This parameter defines the file name of the target file with data using the HTML format. File extension must be **htm** or **html**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION

ADD_MAPPING

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT HTM D:\DataStore\Contacts.HTML CREATE ContactsImport OPTION FIRST_ROW 2
```

```
GATEWAY IMPORT HTM D:\DataStore\Transactions.HTM APPEND TransLog OPTION SHOW_PROGRESS  
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT HTM Customer.HTM APPEND Customer +  
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

1.9.1.3.7 WKQ (QuattroPro)

Syntax:

```
GATEWAY IMPORT WKQ +  
filename.wq1 {APPEND/APPEND_UPDATE/CREATE/REPLACE} +  
<tablename> +  
OPTION SPECIFICATION_FILE_NAME filename +
```

```
| SHEET_INDEX 0 +
| COLUMN_COUNT value +
| KEYS value +
| ADD_MAPPING value +
| AUTONUM value +
| FIRST_ROW n +
| LAST_ROW n +
| SHOW_PROGRESS value +
| MESSAGES value +
| ERROR_FILE EXCEPT.DAT
```

filename.wq1

This parameter defines the file name of the target file with data using the QuattroPro format. File extension must be **wq1**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHEET_INDEX value

This parameter is used to specify a specific sheet of the spreadsheet file, starting with 0.

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT WKQ D:\DataStore\Contacts.WQ1 CREATE ContactsImport OPTION SHEET_INDEX
0|FIRST_ROW 2
```

```
GATEWAY IMPORT WKQ D:\DataStore\Transactions.WQ1 APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT WKQ Customer.WQ1 APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

1.9.1.3.8 XML (Extended Markup Language)**Syntax:**

```
GATEWAY IMPORT XML +
filename.xml {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION SPECIFICATION_FILE_NAME filename +
|COLUMN_COUNT value +
|KEYS value +
|ADD_MAPPING value +
|AUTONUM value +
|FIRST_ROW n +
|LAST_ROW n +
|SHOW_PROGRESS value +
|MESSAGES value +
|ERROR_FILE EXCEPT.DAT +
|FLATTEN_STRUCTURE value +
|PROCESS_ATTRIBUTES value +
|NODE_PATH value
```

filename.xml

This parameter defines the file name of the target file with data using the XML format. File extension must be **xml**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

FLATTEN_STRUCTURE

Specifies the structured records to be flattened, so that the related data appears as one row. Available Parameters: ON / OFF

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

NODE_PATH

Specifies the full path of the target node (e.g. Fantasy\BOOKS). If the PATH is invalid or is not provided, the dialog prompt to select a node is displayed. The NODE_PATH value is case sensitive.

PROCESS_ATTRIBUTES

Specifies to process the XML data where attributes are contained within each element. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same

parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT XML D:\DataStore\Contacts.XML CREATE ContactsImport OPTION NODE_PATH
Level\Contact
```

```
GATEWAY IMPORT XML D:\DataStore\Transactions.XML APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT XML Customer.XML APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

1.9.1.3.9 MDB (MS Access)

Syntax:

```
GATEWAY IMPORT MDB +
filename.ext {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION SPECIFICATION_FILE_NAME filename +
|SOURCE_TABLE tablename +
|COLUMN_COUNT value +
|KEYS value +
|ADD_MAPPING value +
|AUTONUM value +
|FIRST_ROW n +
|LAST_ROW n +
|SHOW_PROGRESS value +
|MESSAGES value +
|ERROR_FILE EXCEPT.DAT
```

filename.ext

This parameter defines the file name of target file with data using the MS Access format. File extension can be **mdb** or **accdb**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION

ADD_MAPPING

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SOURCE_TABLE value

This parameter is used to specify a specific table of the database file.

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT MDB D:\DataStore\Contacts.MDB CREATE ContactsImport OPTION SOURCE_TABLE
NewContacts|ERROR_FILE ImportError.log
```

```
GATEWAY IMPORT MDB D:\DataStore\Transactions.MDB APPEND TransLog OPTION SHOW_PROGRESS
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT MDB Customer.MDB APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

```
GATEWAY IMPORT MDB Transactions.ACCDB APPEND TransLog OPTION SHOW_PROGRESS ON
```

Note:

The Microsoft Access Database Engine 2010 Redistributable must be installed on the computer to import .accdb files.

1.9.1.3.10 JSON (JavaScript Object Notation)

Syntax:

```
GATEWAY IMPORT JSON +  
filename.json {APPEND/APPEND_UPDATE/CREATE/REPLACE} +  
<tablename> +  
OPTION SPECIFICATION_FILE_NAME filename +  
| COLUMN_COUNT value +  
| KEYS value +  
| ADD_MAPPING value +  
| AUTONUM value +  
| FIRST_ROW n +  
| LAST_ROW n +  
| SHOW_PROGRESS value +  
| MESSAGES value +  
| ERROR_FILE EXCEPT.DAT +  
| NODE_PATH value
```

filename.json

This parameter defines the file name of the target file with data using the JSON format. File extension must be **json**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

NODE_PATH

Specifies the full path of the target node (e.g. Fantasy\BOOKS). The NODE_PATH parameter may be set to the root node path with a backslash character (meaning the root). The change is applicable to JSON files where only the data row items are in the file. If the PATH is invalid or is not provided, the dialog prompt is displayed to select a node. The NODE_PATH value is case sensitive.

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The .RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT JSON D:\DataStore\Contacts.JSON CREATE ContactsImport OPTION NODE_PATH
Level\Contact
```

```
GATEWAY IMPORT JSON D:\DataStore\Transactions.JSON APPEND TransLog OPTION
SHOW_PROGRESS OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT JSON Customer.JSON APPEND Customer +
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

```
GATEWAY IMPORT JSON CustResponse.json REPLACE CustResponse OPTION FIRST_ROW 1|
NODE_PATH \
```

1.9.1.3.11 ODS (OpenOffice Spreadsheet)

Syntax:

```
GATEWAY IMPORT ODS +
filename.ext {APPEND/APPEND_UPDATE/CREATE/REPLACE} +
<tablename> +
OPTION ADD_MAPPING value +
|SPECIFICATION_FILE_NAME filename +
|COLUMN_COUNT value +
|KEYS value +
|AUTONUM value +
|FIRST_ROW n +
|LAST_ROW n +
|SHOW_PROGRESS value +
```

```
|MESSAGES value +  
|ERROR_FILE EXCEPT.DAT
```

filename.ods

This parameter defines the file name of the target file with data using the OpenOffice Spreadsheet format. The file extension must be **ods**.

APPEND

Adds the imported data to the end of the table or view specified in *tblview*.

APPEND_UPDATE

Updates the record if it exists in the *tblview* destination, otherwise the record is added. In order to specify a matching row for the UPDATE comparison, use the KEYS parameter.

CREATE

Creates a destination table with the same name as the imported file.

REPLACE

Replaces the rows in the table or view specified in *tblview* with the imported data.

tblview

Specifies the name of a table or view for import.

OPTION**ADD_MAPPING**

This parameter assigns a source column to a destination table column, rather than relying on column position.

AUTONUM

This parameter specifies whether the data imported will follow the auto-numbering formula for the table. Available Parameters: ON / OFF

COLUMN_COUNT

This parameter specifies the number of columns to be imported from file. COLUMN_COUNT is only supported when using CREATE in the GATEWAY syntax.

ERROR_FILE

Specifies a file to capture errors encountered during the import process.

FIRST_ROW

This parameter specifies the first row(s) to be used within the file that is being imported into the table.

KEYS

This parameter specifies the matching row to compare for table updates with the APPEND_UPDATE import mode. For multiple key fields to locate a matching row, use a comma separated list of field names.

LAST_ROW

This parameter specifies the last row to be used within the file that is being imported into the table.

MESSAGES

If ON, during import process some messages will be displayed such as -ERROR- messages, if any. If OFF, the entire import process will be in "silent" mode. Available Parameters: ON / OFF

SHOW_PROGRESS

Indicates whether to display status dialog during the import process. Set SHOW_PROGRESS to OFF if you wish to hide the animated status dialog during the import process. Available Parameters: ON / OFF

SPECIFICATION_FILE_NAME

This parameter is used to load a prepared specification with all pre-defined settings for importing. The import must be performed once using the graphic user interface and saved using the "Specifications.." button. The specifications will be stored in a file with the .RGW file extension. The

.RGW file acts as a set of default values. Any value in the .RGW file will be overwritten by the same parameter specifically assigned by an OPTION parameter. Since the input file name is required in the GATEWAY command, the input file name in the .RGW will always be overwritten.

Examples:

```
GATEWAY IMPORT ODS D:\DataStore\Contacts.ods CREATE ContactsImport OPTION FIRST_ROW 3
```

```
GATEWAY IMPORT ODS D:\DataStore\Transactions.ods APPEND TransLog OPTION SHOW_PROGRESS  
OFF|MESSAGES OFF
```

--This command will only append cell B, cell C and cell D data into the Company, CustAddress and CustState columns of the Customer table.

```
GATEWAY IMPORT ODS Customer.ods APPEND Customer +  
OPTION ADD_MAPPING Company=B|ADD_MAPPING CustAddress=C|ADD_MAPPING CustState=D
```

1.9.1.4 Import Dialog

The File Gateway utility can be launched with the available import options preselected, allowing for easier file transfers for end users. The **OPTION** keyword must immediately follow the IMPORT keyword in order for the default options to be assigned.

```
GATEWAY IMPORT OPTION <default_parameter> <value>
```

Default Parameter	Value
DEFAULT_EXISTING_TABLE	value
ONLY_TABLE	value
DEFAULT_NEW_TABLE	value
DEFAULT_FORMAT	DB, DBF, TXT, CSV, DAT, PRN, TAB, ASC, HTM, HTML, XLS, XLSX, XLSW, WK1, WK2, WR1, WQ1, XML, MDB, ACCDB, JSON, ODS
DEFAULT_FILENAME	value
DEFAULT_CREATE_TEMPORARY	ON/OFF
DEFAULT_PASSWORD	value
DEFAULT_SHOW_PROGRESS	ON/OFF
DEFAULT_SHOW_MESSAGES	ON/OFF
QUALIFIER	PIPE, <character>
SEPARATOR	COMMA, TAB, SEMICOLON, SPACE, PIPE, NONE, <character>
REC_SEP	CR, LF, CRLF, <character(s)>

Example 01:

-- launches the Gateway Utility with only the Import tab displayed

```
GATEWAY IMPORT
```

Example 02:

-- launches the Gateway Utility Import tab with Customer preselected as the default table, and prevents the selection of other tables

```
GATEWAY IMPORT OPTION DEFAULT_EXISTING_TABLE Customer +  
|ONLY_TABLE Customer
```

Example 03:

-- launches the Gateway Utility Import tab with NewLeads specified as the new table

```
GATEWAY IMPORT OPTION DEFAULT_NEW_TABLE NewLeads
```

Example 04:

-- launches the Gateway Utility Import tab with NewLeads specified as the new table with XLS as the file format

```
GATEWAY IMPORT OPTION DEFAULT_NEW_TABLE NewLeads +
```

```
|DEFAULT_FORMAT XLS
```

Example 05:

-- launches the Gateway Utility Import tab with ACHTrans as the new table, ASC as the file format, and ACH_Import.asc as the file name

```
GATEWAY IMPORT OPTION DEFAULT_NEW_TABLE ACHTrans +
|DEFAULT_FORMAT ASC +
|DEFAULT_FILENAME ACH_Import.asc
```

Example 06:

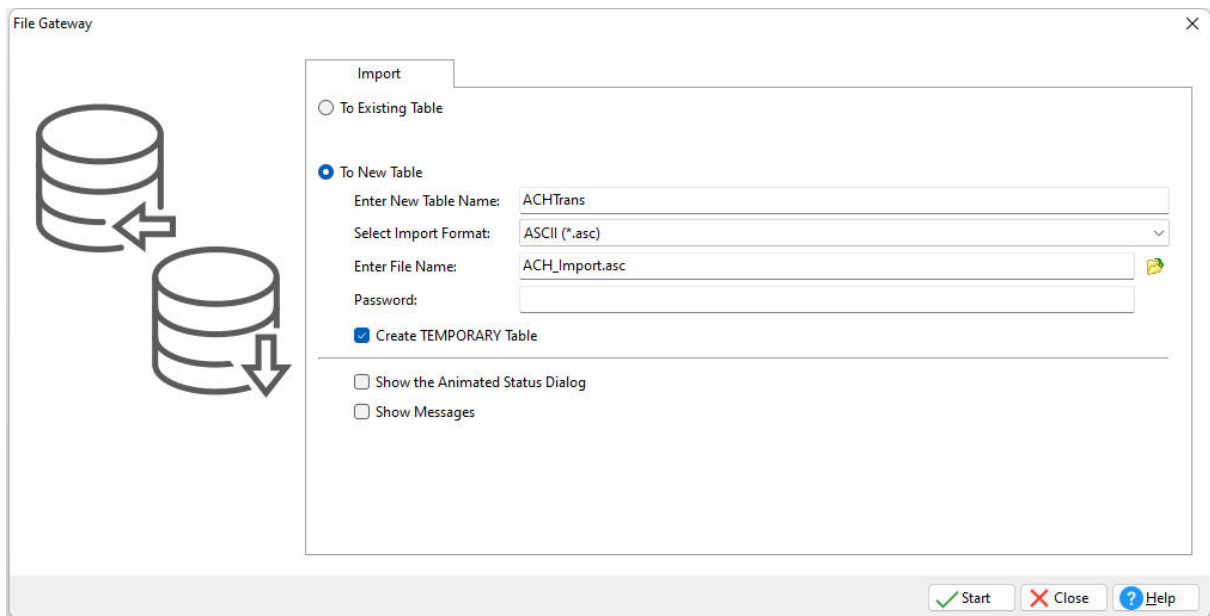
-- launches the Gateway Utility Import tab with Questionnaire specified as the default table, CSV as the file format, and the progress and messages set to display

```
GATEWAY IMPORT OPTION DEFAULT_EXISTING_TABLE Questionnaire +
|DEFAULT_FORMAT CSV +
|DEFAULT_FILENAME Questionnaire.csv +
|DEFAULT_SHOW_PROGRESS ON +
|DEFAULT_SHOW_MESSAGES ON
```

Example 07:

-- launches the Gateway Utility Import tab with ACHTrans as the new "temporary" table, ASC as the file format, ACH_Import.asc as the file name, and the progress and messages set to not display

```
GATEWAY IMPORT OPTION DEFAULT_NEW_TABLE ACHTrans +
|DEFAULT_FORMAT ASC +
|DEFAULT_FILENAME ACH_Import.asc +
|DEFAULT_CREATE_TEMPORARY ON +
|DEFAULT_SHOW_PROGRESS OFF +
|DEFAULT_SHOW_MESSAGES OFF
```



1.9.2 GET

Retrieves a stored procedure.

```
GET [LOCK] procname TO filename
```

Options

filename

The name of the ASCII text format file the stored procedure is placed in.

LOCK

Locks the procedure so it cannot be locked or unlocked by another user. When a procedure is locked, only the user placing the lock can replace the procedure. The [NAME](#) setting is used for identification of the user.

procname

The name of the procedure to retrieve.

About the GET Command

The GET command is used to read a stored procedure from the database into an ASCII file. If the LOCK option is used with the GET command, the procedure cannot be replaced by using the [PUT](#) command.

Rows are copied, not removed, from the source.

Example

The following command retrieves the SetOrderID procedure and places it into a file name SetOrdID.PRO.

```
GET SetOrderID TO SetOrdID.PRO
```

The following series of commands will retrieve the CreateTempTabs procedure and place it into a file name TempTabs.PRO, then Edit the file, and finally replace the stored procedure from the file with an updated version.

```
GET CreateTempTabs TO TempTabs.PRO
RBE TempTabs.PRO
PUT TempTabs.PRO AS CreateTempTabs
```

Note: The PRO file extension is not required by R:BASE, but is provided as the default when unloading a stored procedure from the Database Explorer and is recognized within the R:BASE Editor.

1.9.3 GETPROPERTY

Use the GETPROPERTY command to capture the current properties of forms, form controls, applications, reports, report controls, labels, and label controls. There is also a GETPROPERTY function.

GETPROPERTY	ComponentID	ControlProperty	VarName
	RBASE_FORM	FormProperty	VarName
	APPLICATION	ApplicationProperty	VarName
	RBA_FORM	ApplicationFormProperty	VarName
	REPORT	ReportProperty	VarName
	TABLE_COLUMN	ColumnProperty	VarName
	BLOB_EDITOR	BlobEditorProperty	VarName
	LOADWINDOW	WindowProperty	VarName

Options

APPLICATION ApplicationProperty

Specifies to capture an application property such as the CAPS Lock status, or the application title

TABLE_COLUMN ColumnProperty

Specifies to capture a table's column value

BLOB_EDITOR BlobEditorProperty

Specifies to capture a status of the BLOB Editor utility

Component ID

Specifies the unique identifier assigned to a control (e.g. DB Edit). The Component ID within the GETPROPERTY function must match the unique identifier listed in the control properties. The "Component ID" field is located in the Object Properties for all forms, reports, labels and controls. A "Component ID" option is also located in the speed menu list when you right click on a report/label control.

ControlProperty

The name of the form/report/label object's property you wish to get the value for

LOADWINDOW WindowProperty

Specifies to capture the load window property such as the caption or title. The LOADWINDOW parameter is only supported in R:Compiler

RBA_FORM ApplicationFormProperty

Specifies to capture an application form property such as the current theme, the enabled status for an action, or a caption

RBASE_FORM FormProperty

Specifies to capture a form property such as height, width, etc.

REPORT ReportProperty

Specifies to capture a report property such as the report name, the total page count, or the number of tables used in the report

VarName

The variable name with resulting value as text

Tips:

- The properties of any form control objects can be captured with assigned Component ID values ONLY.
- Always use the current [QUOTES](#) character around the variable name.
- The resulting variable value will be TEXT. To manipulate any values to INTEGER, use the INT, FLOAT, or other conversion function.
- To change any property of a form control after the form is displayed, use the [PROPERTY](#) command(s) within the "On After Start" EEP section of the Form Properties.

Notes:

- The complete list of GETPROPERTY command parameters are available within the **FormProperties.pdf** and **FormProperties_TOC.pdf** PDF documents, which are provided within the R:BASE program directory and are available for download at the R:BASE Technologies Support page: <http://www.rbase.com/support/>
- An optional syntax builder add-on product called [R:Docs](#), containing all PROPERTY/GETPROPERTY commands and parameters constructed in an R:BASE application, is also available. Please contact the R:BASE Technologies Sales Team at sales@rbase.com if you wish to acquire an annual subscription to the stand-alone R:Docs database and application.

Examples:

Example 01.

Gets the current color of any control. The resulting variable vColor will contain the current color name of that form control, such as: BLUE

```
GETPROPERTY ComponentID COLOR 'vColor'
```

Example 02.

Gets the current alignment of any control. The resulting variable vAlignment will contain the current alignment of that form control, such as: LEFT

```
GETPROPERTY ComponentID ALIGNMENT 'vAlignment'
```

Example 03.

Gets the current status of animated GIF control. The resulting variable vGIFAct will contain the current Animated option of Animated GIF Control property, such as: TRUE

```
GETPROPERTY ComponentID ANIMATE 'vGIFAct'
```

Example 04.

Gets the current caption value of any control. The resulting variable vCaption will contain the current text caption of that form control.

```
GETPROPERTY ComponentID CAPTION 'vCaption'
```

Example 05.

Gets the width of form. The resulting variable vWidth will contain the form width.

```
GETPROPERTY RBASE_FORM WIDTH 'vWidth'
```

Example 06.

Gets the height of form. The resulting variable vHeight will contain the form height.

```
GETPROPERTY RBASE_FORM HEIGHT 'vHeight'
```

Example 07.

Gets the left coordinates of form. The resulting variable vLeft will contain the form left coordinates.

```
GETPROPERTY RBASE_FORM LEFT 'vLeft'
```

Example 08.

Gets the right coordinates of form. The resulting variable vRight will contain the form right coordinates.

```
GETPROPERTY RBASE_FORM RIGHT 'vRight'
```

Example 09.

After capturing the Left and Top coordinates, use the INT Function to convert the TEXT value to an INTEGER value.

```
GETPROPERTY RBASE_FORM LEFT 'vLeft'
```

```
GETPROPERTY RBASE_FORM TOP 'vTop'
```

```
SET VAR vLeftPixel INTEGER = (INT(.vLeft))
```

```
SET VAR vTopPixel INTEGER = (INT(.vTop))
```

1.9.4 GOTO

Use the GOTO command in a program to pass control to the commands following the [LABEL](#) command.

GOTO lblname

Option

lblname

Specifies a 1 to 18 character name that labels a line to skip to when a GOTO command is executed in a command or procedure file.

About the GOTO Command

You should limit the use of the GOTO command because GOTO runs more slowly than other R:BASE control structures. Instead, when possible, use a [WHILE](#) loop, [SWITCH](#) structure, or [IF](#) structure to build the command-file logic. Never use GOTO to exit from a WHILE loop or SWITCH structure.

Using the LABEL Command with GOTO

GOTO must have a corresponding [LABEL](#) command within the same command block or file. The LABEL command may precede or follow GOTO in the same command file or, in a procedure file, within the same command block.

You can use a variable containing the name of the label instead of using the specific label name in the GOTO command. To do this, you must use a dot or ampersand variable to tell R:BASE to use the contents of a variable, rather than the variable name as the label name.

Examples

The following example uses a dotted variable containing the name of the label instead of using the specific label name in the GOTO command. If the variable was not a dotted variable, R:BASE would search for a label named *vlabel*. Because it is a dotted variable, R:BASE looks for the correct label name *label1*.

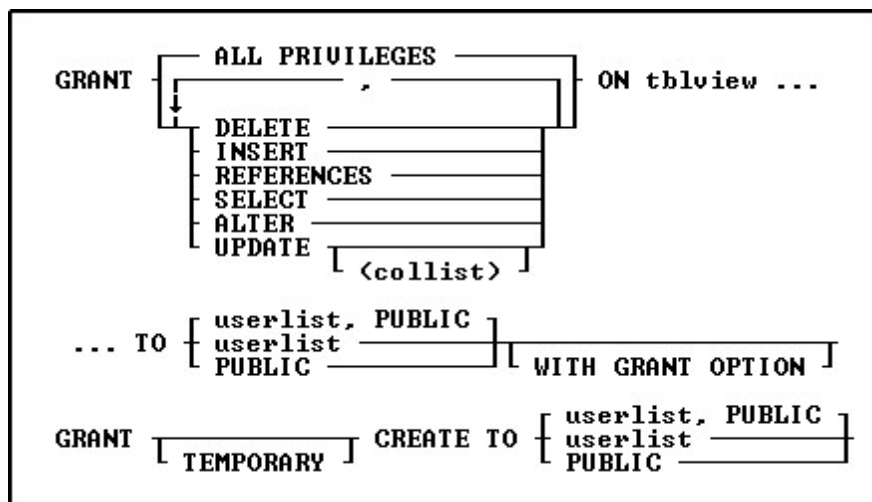
```
SET VARIABLE vlabel = 'label1'
GOTO .vlabel
```

The GOTO *lexit* command in the following example causes the commands following the ENDIF command to be skipped and the [QUIT TO](#) command to be run. The only way the commands between the IF structure and LABEL *lexit* command would be executed would be if the value of *v1* is not greater than the value of *.v2*.

```
IF v1 > .v2 THEN
  GOTO lexit
ENDIF
.
.
.
LABEL lexit
QUIT TO caller
```

1.9.5 GRANT

Use the GRANT command to assign privileges to users of a table or view.



Options

' Indicates that this part of the command is repeatable.

ALL PRIVILEGES

Grants all user privileges on the specified table, or on a view that can be updated.

ALTER

Grants permission to alter specific tables.

CREATE TO

Grants permission to users to create tables using the [CREATE TABLE](#) command. Users who have been granted permission to use this command have all privileges on the tables they create, including the WITH GRANT OPTION. However, users do not have privileges on any other tables in the database unless they are specifically granted permission by the owner.

DELETE

Grants permission to remove rows from the specified table or from a view that can be updated.

INSERT

Grants permission to add rows to the specified table or to a view that can be updated.

ON tblview

Specifies a table or view.

PUBLIC

Grants specified user privileges to all users.

REFERENCES

Grants permission to create a table with a foreign key that references a table with a primary key.

SELECT

Grants permission to display or print data for the specified table or view.

TEMPORARY

Grants permission to users to create temporary tables. Users who have been granted permission to use this command have all privileges on the temporary tables they create, including the WITH GRANT OPTION. However, users do not have privileges on any other tables in the database unless they are specifically granted permission by the owner.

UPDATE (collist)

Grants permission to change the values of columns in the specified table or a view that cannot be updated. If you do not include the optional (*collist*), the user can update all columns in the table. If you list columns, the user can update only the specified columns.

userlist

Grants specified user privileges to listed users. You must separate user identifiers with a comma (or the current delimiter). For a value with spaces, the userid must be enclosed in quotes.

userlist, PUBLIC

Grants specified user privileges to listed users and PUBLIC. Users in *userlist* can retain their user privileges if user privileges granted to PUBLIC are revoked. If, for example, *Ralph*, *Sam*, *Jane*, and PUBLIC have been granted certain user privileges, revoking those privileges from PUBLIC would not affect the three listed users. You must separate the user identifier with a comma (or the current delimiter). For a value with spaces, the userid must be enclosed in quotes.

WITH GRANT OPTION

Allows the specified users to pass the granted user privileges to other users. When you use the [LIST ACCESS](#) command, an asterisk is displayed in front of the user privilege to show a user can grant the assigned user privilege to others; for example, *SELECT means a user has permission to display or print data for specified tables or views, and can grant SELECT rights to other users.

About the GRANT Command

As the database owner, you must first set your own user identifier. After setting your user identifier, you can assign privileges to other users for the tables or views in your database. You must specifically grant privileges to other users. You can assign privileges for a table to individual users, to PUBLIC, or to both. Each user can have a different set of user privileges for the same table, and you can grant a user the right to grant user privileges to others. You can set your user identifier with the [RENAME OWNER](#) command and assign user privileges to other users by using the GRANT command.

In R:BASE for Windows you can also set your user identifier by choosing **Utilities: Set User ID and Password**. To assign user privileges to other users, choose the **User Privileges** option from the **Utilities** menu.

Granting User Privileges

You grant user privileges or access rights on tables or views, however, UPDATE rights must be granted at the column level and CREATE rights must be granted at the database level. If you assign more than one user privilege in a single GRANT command, separate the user privileges with a comma (or the current delimiter).

You can grant the following user privileges: ALL PRIVILEGES, ALTER, CREATE, DELETE, INSERT, REFERENCES, SELECT, and UPDATE; however, you can grant only the SELECT user privilege on views that cannot be updated.

Using User Identifiers and Passwords

A user identifier can be any unique string that uniquely identifies a user to the system. A user identifier can be of 128 characters (or less). To maximize security, create user identifiers that are difficult to guess—such as a random string of letters and numbers. Users can assign passwords to their user identifiers for an added level of security. For information about users assigning passwords see [SET USER](#).

In a database where users have been assigned rights, printing reports requires one of these conditions:

- A user has been granted SELECT privileges on the driving table or view and any look-up tables.
- A user has been granted SELECT privileges or ALL PRIVILEGES on all tables used for the report.
- PUBLIC has been granted SELECT privileges on the driving table or view.
- PUBLIC has been granted SELECT privileges or ALL PRIVILEGES on all tables used for the report.

The only exception to this system of assigning rights is password-protected forms. Passwords assigned to forms, override user privileges assigned with the GRANT command. If a form has not been assigned a password, the user privileges you granted to the tables associated with the form are in effect.

Once R:BASE determines that a user can have access to a password-protected form, R:BASE does not verify user privileges on the underlying tables. Therefore, access to a password-protected form overrides table-level user privileges, making it possible for a user who does not have user privileges on a table to modify the information in that table.

Creating New Tables

To create new tables in a database, a user must be assigned the CREATE user privilege. R:BASE assigns all user privileges to the user for all tables created, including the GRANT user privilege.

A user must be assigned the SELECT user privilege to create a new table from existing tables using the [INTERSECT](#), [JOIN](#), [PROJECT](#), [SUBTRACT](#), or [UNION](#) commands. R:BASE assigns users who use these commands all user privileges on the new table. These user privileges do not include the GRANT user privilege.

Creating Views

[CREATE VIEW](#) also requires the SELECT user privilege on the existing tables. R:BASE assigns users who create views the same user privileges they have on the source table. For views that cannot be updated, R:BASE only assigns users the SELECT user privilege.

Command Authorization Requirements

The following three tables list R:BASE commands and the user privileges they require.

R:BASE Commands that Require the SELECT Access Right

Command	SELECT Access Right on...
BACKUP DATA	Table
BROWSE	Table
COMPUTE	Table
CREATE VIEW	Component tables

CROSSTAB	Table
DECLARE CURSOR	Table
FETCH	Table
INTERSECT	Table 1 and table 2
JOIN	Table 1 and table 2
OPEN CURSOR	Table
PRINT	Driving table/view, and any look-up tables
PROJECT	Table 1
QUERY	Table
SELECT	Table
SET VARIABLE *	Table
SET STATICVAR *	Table
SUBTRACT	Table 1 and table 2
TALLY	Table
UNION	Table 1 and table 2
UNLOAD DATA	Table

* SET VARIABLE and SET STATICVAR commands require the SELECT user privilege only when the value of the variable is derived from a column.

R:BASE Commands that Require the UPDATE User Privilege

Command	UPDATE User Privilege on...
CREATE INDEX	Column
EDIT ALL	Table. All columns in the table are displayed if you have SELECT permission on the table.
QUERY	Column
UPDATE	Column list

R:BASE Commands that Require Other User Privileges

Command	User Privilege	Access on...
DELETE	DELETE	Table or single-table view.
EDIT USING	UPDATE, SELECT, DELETE, ALL PRIVILEGES	If the form is protected by a password at either the read or write level, the password on the form is the overriding user privilege. If the form is not protected by a password, the owner of the database must grant the specific user privileges on the tables used in the form.
ENTER	INSERT, ALL PRIVILEGES	If the form is protected by a password at either the read or write level, the password on the form is the overriding user privilege. If the form is not protected by a password, the owner of the database must grant the specific user privileges on the tables used in the form.
INSERT	INSERT	Table or single-table view, without calculations.
LIST ALL	Any user privilege	Any user privilege granted allows users to list all tables for which they have user privileges.
LIST COLUMNS	Any user privilege	Any user privilege granted allows users to display columns for which they have user privileges.
LIST CONSTRAINTS	Any user privilege	Any user privilege granted allows users to display constraints for tables for which they have user privileges.
LIST INDEXES	Any user privilege	Any user privilege granted allows users to display indexes for which they have user privileges.
LIST SYS %	SELECT	The SELECT user privilege allows users to view a generalized list of system tables and views.
LIST TABLE SYS %	SELECT	The SELECT user privilege allows users to view a detailed list of system tables and views.
LIST TABLES	Any user privilege	Any user privilege granted allows users to display tables for which they have user privileges.
LIST VIEWS	Any user privilege	Any user privileges allows users to display views for which they have user privileges.

The following table lists the user privileges and the commands that use them. Some commands appear under more than one user privilege.

User Privileges for R:BASE Commands

Access Right	R:BASE Commands that Require The Access Right		
ALTER	ALTER TABLE	AUTONUM	DROP COLUMN
CREATE	ALTER TABLE	DROP	REVOKE ¹
	AUTONUM	GRANT ¹	RULES
	BACKUP ALL	PACK	UNLOAD ALL
	BACKUP STRUCTURE	RELOAD	UNLOAD STRUCTURE
	COMMENT ON	RENAME	
	CREATE TABLE	RESTORE	
Database owner's user identifier	ALTER TABLE	DROP	REVOKE ¹
	AUTONUM	GRANT ¹	RESTORE
	BACKUP ALL	PACK	RULES
	BACKUP STRUCTURE	RELOAD	UNLOAD ALL
	COMMENT ON	RENAME	UNLOAD STRUCTURE
	CREATE TABLE		
DELETE	DELETE	EDIT	LIST ³
Form password ²	EDIT USING ²	ENTER ¹	
INSERT	EDIT	ENTER ²	LIST ³
	EDIT USING	INSERT	LOAD
REFERENCES	ENTER ²	INSERT	UPDATE
	EDIT	LIST ³	
	EDIT USING	LOAD	
SELECT	BACKUP DATA	FETCH	SELECT
	BROWSE	INTERSECT	SET VARIABLE ⁴
	COMPUTE	JOIN	SUBTRACT
	CREATE VIEW	LIST ³	TALLY
	CROSSTAB	PRINT	UNION
	DECLARE CURSOR	PROJECT	UNLOAD DATA
UPDATE	CREATE INDEX	EDIT USING	UPDATE
	EDIT	LIST ³	

1. GRANT and REVOKE do not require the database owner's user identifier for an user privilege that includes GRANT permission.
2. Form passwords override user privileges assigned with the GRANT command. If a form does not have a password, the INSERT, DELETE, SELECT, or UPDATE user privileges are required for the underlying tables.
3. Any user privilege granted allows users to list all tables for which they have user privileges.
4. SET VARIABLE requires the SELECT user privilege only when the value of the variable is derived from a column.

Revoking User Privileges

The database owner can remove user privileges with the REVOKE command. The syntax for the REVOKE command is the same as the syntax for the GRANT command. If you issue the REVOKE ALL PRIVILEGES command without specifying a table, R:BASE revokes all user privileges including ALTER and CREATE.

Examples

The following command grants user privileges to display the view named *SLSView* to a specific user-Jane, and to all users-PUBLIC.

```
GRANT SELECT ON SLSView TO Jane, PUBLIC
```

The following command grants user privileges to add or remove information to or from the *TransMaster* table to any user entering the user identifier *Sam* or *Ralph*.

```
GRANT INSERT, DELETE ON TransMaster TO Sam, Ralph
```

The following command grants user privileges to display and enter information in the *TransMaster* table. Also, the command allows any user entering the user identifier *Jane* to pass the SELECT and INSERT user privileges on to other users.

```
GRANT SELECT, INSERT ON TransMaster TO Jane WITH GRANT OPTION
```

The following command grants the user *Abe*, who is not the database owner, permission to alter the *Customer* table.

```
GRANT ALTER ON Customer TO Abe
```

The following command line grants the user *Abe*, who is not the database owner, permission to create tables.

```
GRANT CREATE TO Abe
```

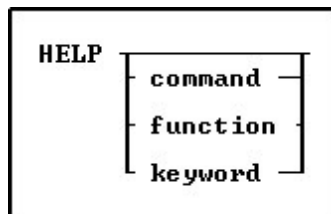
The following command line grants the user *Noah* permission to create temporary tables.

```
GRANT TEMPORARY CREATE TO Noah
```

1.10 H

1.10.1 HELP

Use the HELP command to start the R:BASE online Help.



About the HELP Command

To start the online Help, press [F1].

At the R> Prompt, the Help can be launch by entering "HELP," or "HELP" followed by a command name, function name, or keyword.

Press [Shift] + [F1] to display context-sensitive online Help for the current R:BASE module. Pressing [Shift] + [F1] within the Form Designer will launch the Form Designer Help. Pressing [Shift] + [F1] within the Report Designer will launch the Report Designer Help.

The help file contains all the R:BASE help information and must be located in the same directory as the R:BASE program. All help files have the .CHM extension.

Examples

The following launches the help file and displays the [CHOOSE](#) command.

```
HELP CHOOSE
```

The following launches the help file and displays the ADDYR function.

```
HELP ADDYR
```

The following launches the help file and displays the available icons for the [DIALOG](#) and [PAUSE](#) commands.

HELP ICONS

1.11 I

1.11.1 IF/ENDIF

Use an IF...ENDIF structure in a command file to cause a block of commands to be run when the specified conditions are met.

<pre>IF <i>condlist</i> THEN <i>then-block</i> ENDIF</pre>	<pre>IF <i>condlist</i> THEN <i>then-block</i> ELSE <i>else-block</i> ENDIF</pre>
--	---

Options

condlist

Lists a set of conditions that combine to form a statement that is either true or false. Conditions are combined with the connecting operators AND, OR, AND NOT, and OR NOT.

else-block

Contains one or more R:BASE commands to execute when the conditions specified in *condlist* are false.

then-block

Contains one or more R:BASE commands to execute when the conditions specified in *condlist* are true.

About the IF...ENDIF Command

When the conditions in an IF...ENDIF structure are true, R:BASE runs all the commands between THEN and ELSE, or if the ELSE option is not included, between the THEN and ENDIF.

If you use the ELSE option and the conditions are false, R:BASE runs the block of commands between the ELSE and the ENDIF. If you do not use the ELSE option and the conditions are false, R:BASE runs the command line immediately after ENDIF.

IF...ENDIF structures can be nested with other IF...ENDIF structures.

IF...ENDIF structures can be on a single line in a command file. You cannot put an IF...ENDIF structure on a single line when any of the following occur in a command file:

- The last command in the then-block is [QUIT](#).
- The structure contains an else-block.
- The structure contains a [RUN](#) command.

Using Conditions in an IF...ENDIF Structure

The conditions for an IF...ENDIF structure are listed in the table below.

Condition	Description
<i>varname</i> IS NULL	The value of the variable is null.
<i>varname</i> IS NOT NULL	The value of the variable is not null.
<i>varname</i> CONTAINS ' <i>string</i> '	The variable has a TEXT data type and contains a ' <i>string</i> ' as a substring in the variable value. A ' <i>string</i> ' can contain wildcards.
<i>varname</i> NOT CONTAINS ' <i>string</i> '	The variable has a TEXT data type and a ' <i>string</i> ' is not contained as a substring in the variable value. A ' <i>string</i> ' can contain wildcards.

<i>varname</i> LIKE ' <i>string</i> '	The variable equals a ' <i>string</i> .' A ' <i>string</i> ' can contain wildcards.
<i>varname</i> NOT LIKE ' <i>string</i> '	The variable does not equal the ' <i>string</i> '. A ' <i>string</i> ' can contain wildcards.
<i>varname</i> LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	The variable equals a ' <i>string</i> .' If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>varname</i> NOT LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	The variable does not equal a ' <i>string</i> .' If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>varname</i> BETWEEN <i>value1</i> AND <i>value2</i>	The value of the variable is greater than or equal to <i>value1</i> and less than or equal to <i>value2</i> . The variable and the values must be the same data type.
<i>varname</i> NOT BETWEEN <i>value1</i> AND <i>value2</i>	The value of the variable is less than <i>value1</i> or greater than <i>value2</i> . The variable and the values must be the same data type.
<i>varname</i> IN (<i>valuelist</i>)	The value of the variable is in the value list.
<i>varname</i> NOT IN (<i>valuelist</i>)	The value of the variable is not in the value list.
<i>item1</i> op <i>item2</i>	<i>Item1</i> has the specified relationship to <i>item2</i> . <i>Item1</i> can be a column name, value, or expression; <i>item2</i> can be a column name, value, or expression.

The valid operators (*op*) for the conditions in an IF...ENDIF structure are listed in the table below. Do not use wildcard characters with these operators.

Operator	Description
=	Equals
<=	Less than or equal to
>=	Greater than or equal to
<	Less than
>	Greater than
<>	Not equal

An expression can be substituted for the first variable in each of the conditions. The second variable in the comparison must be dotted so that the value of the variable is used, not the variable name.

Wildcards can be used with LIKE, NOT LIKE, CONTAINS, or NOT CONTAINS (for example, *varname* LIKE '*string*%').

If you use multiple AND and OR operators, you must enter them in the correct order, or use parentheses to obtain the desired result. If [SET AND](#) is on (the default setting) conditions connected by AND are evaluated first; then conditions connected by OR are evaluated.

Examples

The following command lines show three nested IF...ENDIF structures.

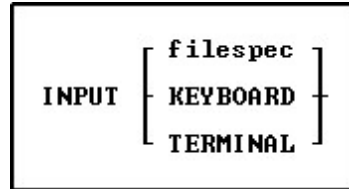
```
IF vqtyord > .vlastqty THEN
  IF vqtyord <> 0 THEN
    IF vchng > 0 AND vbackord IS NOT NULL THEN
      .
      .
      .
    ELSE
      .
      .
      .
    ENDIF
  ENDIF
ENDIF
```


The following example shows an IF...ENDIF structure on a single line.

```
IF v2 = 0 THEN ; SET VARIABLE v1 = (.v1 + 1) ; ENDIF
```

1.11.2 INPUT

Use the INPUT command to make a specified file current input device.



Note: When migrating from DOS to Windows it is recommended that you replace your INPUT commands with the [RUN](#) command.

The INPUT command is used most often to load an ASCII file, created with the [OUTPUT](#) and [UNLOAD](#) commands, back into a database.

The INPUT command also transfers the contents of a backup file generated by [BACKUP](#) or [UNLOAD](#). INPUT executes the commands in the file and loads the data contained in the backup file. Do not use INPUT when the backup file is on more than one floppy disk.

filespec

Specifies an input file. If you specify a command file, it starts after you press [Enter].

KEYBOARD

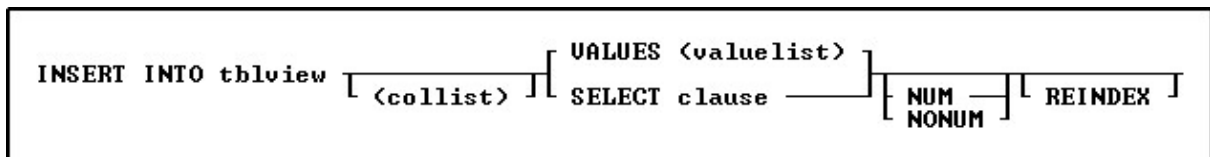
Clears all nesting levels and returns control to the R> Prompt.

TERMINAL

Clears all nesting levels and returns control to the R> Prompt.

1.11.3 INSERT

Use the INSERT command to add data to a table or view without using a data-entry form.



Options

(collist)

Specifies a list of one or more column names, separated by a comma (or the current delimiter). In an SQL command, any column name in the list can be preceded by a table or correlation name and a period (*tblname.colname*).

INTO tblview

Specifies the table or view name (views must be updatable).

NUM

NONUM

NUM specifies that autonumbering columns will be numbered as they are inserted. NONUM turns off autonumbering while inserting, thereby allowing inserting of a specific value for autonumber columns. The default is NUM.

REINDEX

Specified that R:BASE will temporarily add the new rows without updating every non-constraint index row by row, and only updates constraint indexes row by row. When the new rows are done being added, the non-constraint indexes are then rebuilt. The rebuilding of the non-constraint indexes after the rows are added increases the overall speed of the operation.

SELECT clause

Finds values in a table, tables, or view to insert into the table or view specified by the INTO *tblview* option and the columns specified by the *collist* option.

VALUES (vallist)

Specifies a list of values to insert into the table specified by the INTO *tblview* option and the columns specified by the *collist* option. Separate values with a comma or the current delimiter.

For these data types...	Use this format for vallist
All data types except BIT, BITNOTE, LONG VARBIT, and VARBIT	'string' or value
BIT, BITNOTE, LONG VARBIT, LONG VARCHAR, VARBIT, and VARCHAR	['filename.ext'] or ['filename.ext', filetype, offset, length] Note: When you use VARCHAR, the filetype is always TXT. When you use VARBIT, BIT, and BITNOTE, filetype refers to the standard graphical file types.

About the INSERT Command

The INSERT command assigns a default value of null to any column not named in the *collist* unless a default value has been assigned to a column with the [CREATE TABLE](#) or [ALTER TABLE](#) command.

To ensure that rules are checked while adding data with the INSERT command, [SET RULES ON](#) before running the INSERT command.

The setting of the [SET ZERO](#) command affects the calculation of numeric computed columns. To have null values treated as zeros in expressions, set ZERO on. When ZERO is set off, if the value of a column used in an expression is null, the computed value will be null.

You cannot insert values into the table used in the [SELECT](#) clause.

To ensure that data is placed in the intended column, use the following guidelines:

- Do not embed commas within entries for CURRENCY, DOUBLE, INTEGER, NUMERIC, or REAL data types. R:BASE automatically inserts commas and the current currency symbol.
- When values for CURRENCY, DOUBLE, NUMERIC, or REAL or data types are decimal fractions, you must enter the decimal point. When values are whole numbers, R:BASE adds a decimal point for you at the end of the number. R:BASE adds zeros for subunits in whole currency values; For example, using the default currency format, R:BASE loads an entry of 1000 as \$1,000.00.
- When values for NOTE or TEXT data types contain commas, you can either enclose the entries within quotes, or use [SET DELIMIT](#) to change the default delimiter (comma) to another character.
- When values for NOTE or TEXT data types contain single quotes ('), and you are using the default QUOTES character ('), use two single quotes (') in the text string. For example, 'Walter Finnegan's order.'
- When a value you specify for a column is not the same data type as the column's data type, R:BASE displays an error message and you need to re-enter the entire row.
- When values for NOTE or TEXT data types exceed the maximum length of a column, R:BASE truncates the value and adds it to the table. A message is displayed that tells you which row has been truncated.

Inserting an Autonumbered Column

When you use INSERT to add a row, INSERT assigns the next available number to autonumbered columns in the table. Therefore, omit autonumbered columns and their values from a *collist*. Also, if you

use the [SELECT](#) option, omit an autonumbered column from the *collist*. If a value is included for an autonumbered column that was omitted from the column list, R:BASE does not run the command because it cannot identify which column to load.

Inserting a Computed Column

Because a computed column's value is calculated, you cannot insert a new value. Omit computed columns from a *collist* or, if you are adding data to all columns, do not use a *collist* and do not specify a value for the computed column. R:BASE will skip the computed column when the row is inserted.

Examples

In the following example, the *sales* table has three columns, *col1*, *col2*, and *col3*; and *col2* is a computed column. To insert a row, you would only specify values for *col1* and *col3*. In this example, the value for *col1* is 100, and the value for *col3* is 200.

If the expression for *col2* was $(col1 + 200)$, then *col2* would have the value 300 when the row is inserted.

```
INSERT INTO sales VALUES (100, 200)
```

In the following example, a *vallist* adds a new row to the *product* table, filling the *model*, *prodname*, *proddesc*, and *listprice* columns.

```
INSERT INTO product (model, prodname, proddesc, listprice) +
VALUES ('PB3060', 'Portable Advanced PC', 'System-Single +
Drive w/Hard Disk-Portable', 3795)
```

The following command uses a *vallist* with global variables to insert the values from variables *v1*, *v2*, and *v3* into the *bonusrate* table.

```
SET VARIABLE v1 CURRENCY = 50000, v2 CURRENCY = 75000, +
v3 REAL = .10
INSERT INTO bonusrate VALUES (.v1, .v2, .v3)
```

The following example adds rows to *customer* table selected from *temp* table. It adds data into the *company* and *custphone* columns. The columns taken from the *temp* table can have different column names, the data types must be the same, and the order and number of columns in the column list of the source table (designated by *SELECT*) must match the column list of the destination table (designated by *INTO*).

```
INSERT INTO customer (company, custphone) +
SELECT cname, phone FROM temp
```

1.11.4 INTERSECT

Use the *INTERSECT* command to form a new table by combining two tables that have one or more common columns.

```
INTERSECT tblname1 WITH tblname2 FORMING [ TEMPORARY ] tblname3 [ USING collist ]
INMEMORY
```

Options

FORMING tblname3

Specifies the name of the new table.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using *INMEMORY* in place of *TEMPORARY* creates the temporary table in memory, rather than a temporary file.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

tblname1

Specifies the name of the first table to intersect.

USING collist

Specifies which columns to include in the new table. The clause must include at least one common column. You can use the clause to order the columns in the new table and to determine which columns from the original tables will be in the new table.

If you do not include the clause, R:BASE uses all columns from both tables. Columns from *tblname1* provide the order for rows and columns in the new table. R:BASE adds columns from *tblname2* that are not in *tblname1* to the right of the columns from *tblname1*.

The USING clause also determines which common columns are used for comparison. For example, if *firstname* and *lastname* are in both tables but only *lastname* is included in the USING clause, INTERSECT will compare the tables based only on *lastname*.

WITH tblname2

Specifies the name of the second table to intersect.

About the INTERSECT Command

A new table only contains the rows in which there were identical values in each of the common columns in both tables. If the common columns for a row in each table have the same value, R:BASE combines the two rows into one in the new table. If the common columns in each table have multiple matches, R:BASE adds a row for each match in the new table. For example, if *tblname1* has two rows with an identification number of 100 and *tblname2* has three rows with the same identification number, *tblname3* will have six rows, one for each possible combination.

If you combine one or more NOTE columns with the INTERSECT command, R:BASE might truncate the last NOTE column. The maximum total row length is 32,768 characters.

In the new table, R:BASE does not update the value in the row for the autonumbered column. The autonumbered column becomes a regular column.

Including Computed Columns in a New Table

R:BASE uses the current values in computed columns to determine whether the values in common computed columns are identical. To include a computed column in a new table, include each column used in the computed column's definition in the USING clause, placing the columns before the computed column. Without such a USING clause, R:BASE makes the computed column a regular column, and displays a message suggesting you rename the column in the new table. If you do not rename the column, the new table will have a column that has the same name but a different data type from the original table. If you have columns with the same name but different data types, you will not be able use the BACKUP and UNLOAD commands.

Increasing the Speed of INTERSECT

You can increase the speed of the INTERSECT command by creating an index for the common column in *tblname2*. The INTERSECT command does not use an index in *tblname1*. Below are guidelines for indexing columns in *tblname2*:

- If one table has more rows than the other table, place the larger table second (*tblname2*) in the command and create an index for its common column.
- If both tables have many rows, place the table with the shorter row length first (*tblname1*) and index the common column in the second table.
- If the tables have more than one common column, index the common column that has the most unique values in the second table (*tblname2*).

Examples

The following command forms the *models* table by combining the rows from the *component* and *compused* tables that contain identical values in a column common in both tables. Because the USING clause is not included, the *models* table contains all columns from both tables.

```
INTERSECT component WITH compused FORMING models
```

The following command lines create a table containing employee information on sales representatives and their sales bonuses. The *repsales* table only contains the rows that match in *employee* and *salesbonus* tables.

```
INTERSECT employee WITH salesbonus FORMING repsales +
    USING emplname, empid, hiredate, netamount, transdate, bonus, bonuspct
```

1.12 J

1.12.1 JOIN

Use the JOIN command to form a new table by combining rows from two existing tables based on a comparison of the columns you specify.

```
JOIN tblname1 USING colname1 WITH tblname2 USING colname2 ...
... FORMING [ TEMPORARY ] tblname3 [ WHERE compare_op ]
```

Option

FORMING tblname3

Specifies the name of the new table.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using INMEMORY in place of TEMPORARY creates the temporary table in memory, rather than a temporary file.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

tblname1

Specifies the name of the first table to join.

USING colname1

Specifies the columns to be compared; *colname1* is a column in *tblname1*.

USING colname2

Specifies the columns to be compared; *colname2* is a column in *tblname2*.

WHERE compare_op

States the condition used to compare columns. If omitted, the condition is = or EQ (equal). Use only the following comparison operators in a JOIN WHERE clause:

Comparison Operator		Rows are included if...
=	EQ	<i>colname1</i> is equal to <i>colname2</i> (the default)
<>	NE	<i>colname1</i> is not equal to <i>colname2</i>
>	GT	<i>colname1</i> is greater than <i>colname2</i>
>=	GE	<i>colname1</i> is greater than or equal to <i>colname2</i>
<	LT	<i>colname1</i> is less than <i>colname2</i>
<=	LE	<i>colname1</i> is less than or equal to <i>colname2</i>

WITH tblname2

Specifies the name of the second table to join.

About the JOIN Command

JOIN combines all columns from two tables into a new table. R:BASE adds rows to the new table (*tblname3*) that satisfy the comparison condition (*compare_op*) of a column from each of the existing tables (*tblname1* and *tblname2*). You must specify the columns to be compared: *colname1* is a column in *tblname1* and *colname2* is a column in *tblname2*.

The important characteristic of JOIN is the relationship between column values in the two tables. R:BASE compares the values of one column in the first table to the values of another column in the second table based on the operator you specify. The two tables do not need common columns to satisfy the comparison.

Column Names

If any columns in the existing tables have the same name, change the column name in one of the tables before you join them or before you use the new table. (R:BASE creates the new table and gives the two columns the same name.) If you want to use the columns with the same column name in the USING clause, change one of them before you join the tables. The column names can be different, but their data type and size must be the same.

When the comparison between the two specified columns is satisfied, R:BASE creates a row in the new table that is a combination of all columns from the row in *tblname1* and its corresponding row in *tblname2*.

NOTE Columns

If you use JOIN to combine one or more columns that have a NOTE data type, R:BASE, starting with the last column that has a NOTE data type, truncates the column until the row length is within the 32,768-character limit.

Autonumbered Columns

In the new table, R:BASE does not update the values for the autonumbered column. The autonumbered column becomes a regular column.

Improving Performance with Indexing

When a WHERE clause is used, you can speed up JOIN by creating an index for the compared column in *tblname2*. JOIN does not use an index in *tblname1*. If one table has more rows than the other table, name the larger table second in the command and index its compared column. If both tables have many rows, place the table with the shorter row length first, then create an index for the compared column in the second table.

Example

The following command creates a new table named *proddetail* by combining rows from the *prodlocation* and *product* tables. The rows included in *proddetail* are based on a comparison of the values in the *model* column in *prodlocation* and the values in the *model* column in *product*. The WHERE clause specifies that the new table will contain only the rows in which the *model* value equals the *model* value. Because the equal sign (=) is the default operator, you could omit the WHERE =. All the columns in *prodlocation* and *product* are included in *proddetail*.

```
JOIN prodlocation USING model WITH product USING model +  
FORMING proddetail WHERE =
```

1.13 L

1.13.1 LABEL

In a [GOTO](#) or [WHENEVER](#) statement, use the LABEL command to identify the command line to which control should be passed.

LABEL lblname

Option

lblname

Specifies a 1 to 18 character name that labels a line to skip to when a GOTO command is executed in a command or procedure file.

About the LABEL Command

After a command file is run once, R:BASE stores the labels in memory. When the command file is run again, R:BASE retrieves the location of a label when the [GOTO](#) or [WHENEVER](#) is run. However, if the GOTO or WHENEVER command is going to be run only once, place the LABEL command below the GOTO or WHENEVER command because it is more efficient for R:BASE to search downward in the command file for the matching label.

Example

In the following example, if the *vctrvar* variable in the [IF...THEN](#) statement is equal to 999, control passes to the command lines following the LABEL command, which defines the *endproc* label. If *vctrvar* does not equal 999, the command lines following the ELSE statement are performed.

```
IF vctrvar = 999 THEN
    GOTO endproc
ELSE
    *( commands to execute )
ENDIF

.
.
.

LABEL endproc
    *( commands to execute )
```

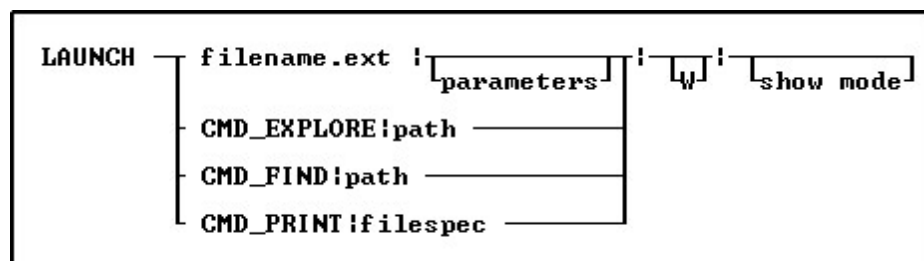
1.13.2 LAST

Use the LAST command in an entry/exit procedure (EEP) to move to the last row.

LAST

1.13.3 LAUNCH

The LAUNCH command invokes or runs executable programs, as well as files associated with programs.



Options

I

The pipe character is needed when a file parameter, W (wait), or show mode is specified. The number of pipes needed depend on the last part of the LAUNCH command being used. If only a file is being launched and no addition portions of the LAUNCH command are being used, the pipe characters are not needed. However, if a file is being launched and only a show mode is used, then all three pipes are required.

CMD_EXPLORE

Displays the Windows Explorer from the specified folder.

CMD_FIND

Initiates a search from the specified folder.

CMD_PRINT

Prints a file to the default printer, or displays a print dialog, based upon file type.

filename.ext

The name of file which has a corresponding association registered with Windows. If the file does not exist, an error will be displayed.

filespec

The name of the file to print. Based upon the file type, the file will be sent directly to the default printer, or will display in a print dialog, If the file does not exist, or if the file is not printable, an error will be displayed.

path

Specifies the directory name.

<parameters>

To specify additional parameters associated to the launched program.

show mode

Specifies how the launched program or operation is displayed. The show modes may not be supported by some programs.

- NORMAL - launches the program in a normal window
- MINIMIZED - launches the program in a minimized window
- MAXIMIZED - launches the program in a maximized window
- HIDE - launches the program, but it is hidden from display

W

To specify the "Wait Until Finished" option.

About the LAUNCH Command

LAUNCH command is used to execute (or LAUNCH) another application or process, from within R:BASE. Based upon the extension for the file name, a corresponding program associated to it within Windows will open when launching the file. For example, launching a text file will invoke Notepad, if the program is associated to the txt file extension.. If a full path is not specified, the current search path is used.

The LAUNCH command allows for command line parameters to be specified that will be recognized by the program being launched. The launched application will start in the same directory as the executable file or in the directory referenced, if specified.

The LAUNCH command also allows you to use a "Wait Until Finished" option to specify whether or not you want R:BASE to sleep while the launched program executes or continue running. If the "Wait Until Finished" parameter "|W", (e.g. pipe and W with no space) is added to the LAUNCH command, R:BASE will sleep until the launched process completes. If the "Wait Until Finished" parameter "|W" is not specified, the launched process is executed in its own thread, and the R:BASE will continue to execute.

The number of pipes needed depend on the last part of the LAUNCH command being used. If only a file is being launched and no addition portions of the LAUNCH command are being used, the pipe characters

are not needed. However, if a file is being launched and only a show mode is used, then all three pipes are required.

Launch Command Features:

- LAUNCH command is supported in EEPs
- If a folder path, file name, or parameter contains spaces, double quotes (") must be added. If a folder/file contains commas, single quotes must be added.
- The LAUNCH command will execute any windows program as long as it is installed and the file extension is associated within the operating system.
- There is no longer a need to use the ZIP OUT command or run batch files, etc.

Examples

Example 01 (Invokes a VBScript to demonstrate how to access Excel using the Windows Script Host):

```
LAUNCH charts.VBS
```

Example 02 (Invokes the PDF reader program on the computer and displays the file):

```
LAUNCH MyDoc.PDF
```

Example 03 (Invokes program associated with text files, e.g. Notepad, and displays the text file in a maximized window. Three pipes are required as the "show mode" parameter is specified.):

```
LAUNCH c:\mydocs\ReadMe.txt |||MAXIMIZED
```

Example 04 (Invokes program associated with text files with the "Wait Until Finished" option for R:BASE to sleep in the background. The text file must be closed in order to give control back to R:BASE. Notice the two pipe characters with no spaces after the file name. The first pipe is listed with no parameters and no space, while the second pipe is for the "Wait Until Finished" option.):

```
LAUNCH ReadMe.txt||W
```

Example 05 (Launches Windows Explorer and displays the contents of the C:\RBTI folder):

```
LAUNCH CMD_EXPLORE|C:\RBTI
```

Example 06 (Launches Windows Explorer and displays the contents of the folder. As the path contains a comma, enclose the string in single quotes.):

```
LAUNCH 'CMD_EXPLORE|C:\Program Files (x86)\Parchment Products, Inc'
```

Example 07 (Initiate a search from the F:\Data folder):

```
LAUNCH CMD_FIND|F:\DATA
```

Example 08 (Prints the ReadMe.txt file to the default printer):

```
LAUNCH CMD_PRINT|C:\ReadMe.txt
```

Example 09 (Invokes MS Word with TestFile.DOC running the MyMacro macro. The command will invoke MS Word with the document using MyMacro as a main window while R:BASE sleeps in the background. Word must be closed in order to give control back to R:BASE. As the folder path contains spaces, double quotes (") are added.):

```
LAUNCH 'winword|"C:\Word Documents\TestFile.DOC" "/myMacro"|W'
```

Example 10 (Prints the OrderForm.pdf document):

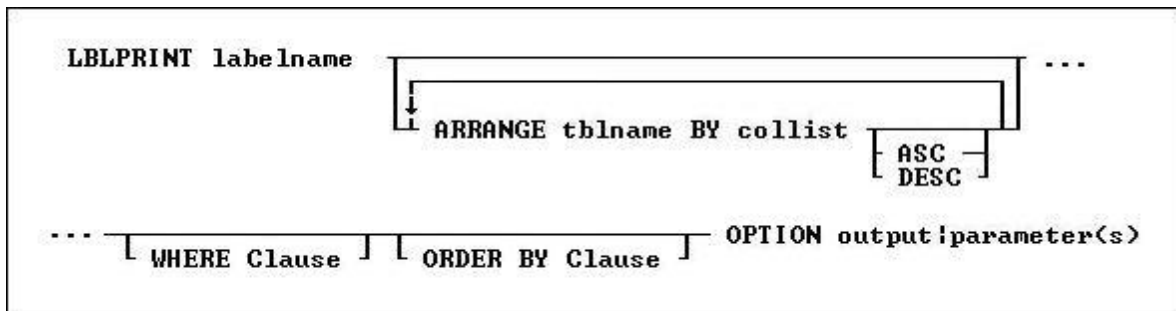
```
LAUNCH CMD_PRINT|N:\Sales\OrderForm.pdf
```

Example 11 (Sends an automated email. The command will launch the default email program on the computer with the To:, Subject, and Body fields populated.):

```
SET VAR vMailTo TEXT = NULL
SET VAR vContID INTEGER = 1010
SELECT ('mailto:' + ContEMail + '?subject=Hello&body=Contact Us') +
INTO vMailTo INDIC IvMailTo FROM Contact WHERE ContID = .vContID
LAUNCH .vMailTo
```

1.13.4 LBLPRINT

Use the LBLPRINT command to send a set of labels to a screen, printer, file, or any combination of destinations.



Options

labelname

Specifies a label name. The label name is limited to 128 characters.

ARRANGE tblname BY collist

Sorts the rows displayed by a report's lower-level tables. Each lower-level table in the report that you want to sort requires its own ARRANGE clause. Now you can specify up to twenty-five tables to ARRANGE and up to five columns in each table on which R:BASE will sort.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

OPTION

specifies output options of the report along with additional parameters for each output such as:

- [SCREEN](#)
- [PRINTER](#)
- [BMP](#)
- [DIF](#)
- [EMF](#)
- [ETXT](#) (Report Emulation Text)
- [GIF](#)
- [HTML](#)
- [JPG](#)
- [LOTUS](#)
- [PDF](#)
- [PNG](#)
- [QPRO](#)
- [RTF](#)
- [SVG](#)
- [SYLK](#)
- [TIFF](#)

- [TXT](#)
- [WMF](#)
- [XHTML](#)
- [XLS](#)
- [XLS2003](#)
- [XLSX](#)

1.13.4.1 SCREEN

The SCREEN output option displays the label in a Print Preview window.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SCREEN
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
WINDOW_STATE	MAXIMIZED MINIMIZED NORMAL	Specifies the state of preview window. By default, the window state is normalized.
TOOLBAR_VISIBLE	TRUE FALSE	Specifies the option to display preview toolbar with all buttons such as, Print, Whole Page, Page Width, 100%, Zoom Level, First, Page Previous Page, Next Page, Last Page and Cancel/Close.
BUTTONBAR_VISIBLE	TRUE FALSE	Specifies the option to display button bar with Whole Page, Page Width and 100%. If set to FALSE, all three buttons (Whole Page, Page Width and 100%) will not be visible.
BTNFIND_VISIBLE	TRUE FALSE	Specifies the option to display the Find Text button on the toolbar
BTNOUTLINE_VISIBLE	TRUE FALSE	Specifies the option to display the Report Outline button on the toolbar
BTNPRINT_VISIBLE	TRUE FALSE	Specifies the option to display Printer button on the toolbar
BTNWHOLEPAGE_VISIBLE	TRUE FALSE	Specifies the option to display Whole Page button on the toolbar.
BTNPAGEWIDTH_VISIBLE	TRUE FALSE	Specifies the option to display Page Width button on the toolbar.
BTNP100PERCENT_VISIBLE	TRUE FALSE	Specifies the option to display 100% button on the toolbar.
ZOOMPERCENT_VISIBLE	TRUE FALSE	Specifies the option to display Zoom Level button on the toolbar.
BTNFIRST_VISIBLE	TRUE FALSE	Specifies the option to display First Page button on the toolbar.
BTNPRIOR_VISIBLE	TRUE FALSE	Specifies the option to display Previous Page button on the toolbar.
PAGENO_VISIBLE	TRUE FALSE	Specifies the option to display Page Number button on the toolbar.
BTNNEXT_VISIBLE	TRUE FALSE	Specifies the option to display Next Page button on the toolbar
BTNLAST_VISIBLE	TRUE FALSE	Specifies the option to display Last Page button on the toolbar.
BTNCANCEL_VISIBLE	TRUE FALSE	Specifies the option to display Cancel/Close button on the toolbar.
ZOOM_TYPE	PAGE_WIDTH WHOLE_PAGE 100_PERCENT PERCENTAGE	Specifies the zoom type. If PERCENTAGE is specified, the ZOOMPERCENT parameter must also be used so the percentage value is provided.
ZOOMPERCENT	value	Specifies the zoom level percentage, such as 70, 100, 120, 200, 250, etc. The ZOOM_TYPE value must be

		defined as PERCENTAGE in order to specify the ZOOMPERCENT value. See examples below.
PREVIEW_CAPTION	value	Specifies the caption for the preview window
PRINTER_NAME	value	Specifies the name of the default selection in the printers drop down of the print dialog, when the Printer button is visible
MDI		Specifies the option to display the preview window in a modeless window, to access other windows without closing it first.
LEFT	value	Specifies the left coordinate of the print preview window
TOP	value	Specifies the top coordinate of the print preview window
WIDTH	value	Specifies the width of the print preview window
HEIGHT	value	Specifies the height of the print preview window
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10- 12,20	Determines the pages in which the Page Style will appear on.
THEMENAME		Specifies one of 86 pre-defined Themes, or a custom Theme loaded into R:BASE. See Themes.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Examples:

-- Example 01 (Typical Print Preview on SCREEN with Toolbar):

```
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED+
|PREVIEW_CAPTION 'Typical Print Preview'
```

-- Example 02 (Typical Print Preview on SCREEN with zoom type as page width):

```
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|ZOOM_TYPE PAGE_WIDTH+
|PREVIEW_CAPTION 'Print Preview Using the Page Width Display'
```

-- Example 03 (Typical Print Preview on SCREEN with zoom type as whole page):

```
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|ZOOM_TYPE WHOLE_PAGE+
```

```
|PREVIEW_CAPTION 'Print Preview Using the Whole Page Display'

-- Example 04 (Typical Print Preview on SCREEN with zoom type as 100%):
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|ZOOM_TYPE 100_PERCENT+
|PREVIEW_CAPTION 'Print Preview Using the Percentage Display (100%)'

-- Example 05 (Typical Print Preview on SCREEN with zoom type as 75%):
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|ZOOM_TYPE PERCENTAGE|ZOOMPERCENT 75 +
|PREVIEW_CAPTION 'Print Preview Using the Percentage Display (75%)'

-- Example 06 (Print Preview Tool Bar without Without the Page Display Buttons):
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNWHOLEPAGE_VISIBLE FALSE+
|BTNPAGEWIDTH_VISIBLE FALSE+
|BTN100PERCENT_VISIBLE FALSE+
|ZOOMPERCENT_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Page Display Buttons'

-- Example 07 (Print Preview Tool Bar without the Navigation Buttons):
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNFIRST_VISIBLE FALSE+
|BTNPRIOR_VISIBLE FALSE+
|PAGENO_VISIBLE FALSE+
|BTNNEXT_VISIBLE FALSE+
|BTNLAST_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Navigation Buttons'

-- Example 08 (Print Preview Tool Bar without Printer Button):
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNPRINT_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Print Button'

-- Example 09 (Print Preview Tool Bar without Cancel/Close Button)
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNCANCEL_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Cancel/Close Button'

-- Example 10 (Print Preview without the entire Tool Bar)
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|TOOLBAR_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Entire Toolbar'

-- Example 11 (Print Preview with a MDI window using specified coordinates)
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SCREEN MDI|LEFT 20|TOP 20|WIDTH 800|HEIGHT 500
```

1.13.4.2 PRINTER

The PRINTER output option sends the label to a printer.

Syntax:

**LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION
PRINTER**

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TRAY		Specifies the name of the bin (paper tray) containing the paper on which the report will be printed. A common use of this property is to set it to Manual Feed on mailing label reports. That way, you can provide a means to load the special label paper without manually setting the printer.
COLLATION	ON OFF	Indicates whether multiple copies will print in pre-sorted order
COPIES	value	Specifies the number of copies to print. If the COLLATION property is set to ON, the copies will print in sorted order.
DOCUMENT_NAME	value	When a report is printing, this is the name of the document as it should appear when listed in the Print Manager and on network header pages. This value can also be displayed in reports using the DocumentName type of the System Variable control.
DUPLEX	NONE HORIZONTAL VERTICAL	Indicates what type of two-sided printing should occur. Not all printers support duplex printing. Horizontal duplex printing is performed across the page, from left to right, so the text will read as in a book. Vertical duplex printing is performed down the page, from top to bottom, so the text will read as a tablet.
ORIENTATION	PORTRAIT LANDSCAPE	Determines whether a report prints vertically (PORTRAIT) or horizontally (LANDSCAPE) on a page.
PAPER_NAME	value	The name of the paper on which the report will be printed. When you specify a value for the PAPER_NAME, the width and height are automatically set to the proper values. For example, if "Legal" is specified, the width will be set to 8.5" and the height will be set to 14".
PAPER_WIDTH		The value of the PAPER_WIDTH is always expressed in the units specified by the report.
PAPER_HEIGHT		The value of the PAPER_HEIGHT is always expressed in the units specified by the report.
PRINTER_NAME		The name of the printer that will be used to print the report.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING	ALL FIRST LAST Page List e.g. 1,4-6,10- 12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).

MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Examples:

--Example 01 (To print Customer List on your default Windows printer):

```
LBLPRINT CustShippingLabel OPTION PRINTER
```

--Example 02 (To print Customer List on your network printer):

```
LBLPRINT CustShippingLabel +
OPTION PRINTER +
|COPIES 1 +
|ORIENTATION PORTRAIT +
|PRINTER_NAME \\MyServer\hp LaserJet 1320 PCL 6
```

--Example 03 (To print Customer Shipping Labels on your label printer):

```
LBLPRINT CustShippingLabel WHERE CustID = 127 OPTION PRINTER +
|COPIES 2 +
|PRINTER_NAME DYMO LabelWriter 2 +
|ORIENTATION LANDSCAPE
```

--Example 04 (To print first page of Customer List report only)

```
LBLPRINT CustShippingLabel OPTION PRINTER|PAGE_SETTINGS FIRST
```

--Example 05 (To print last page of Customer List Report only)

```
LBLPRINT CustShippingLabel OPTION PRINTER|PAGE_SETTINGS LAST
```

--Example 06 (To PRINT all pages)

```
LBLPRINT CustShippingLabel OPTION PRINTER|PAGE_SETTINGS ALL
```

--Example 07 (To print pages 3, 5, and 8 of your big report)

```
LBLPRINT CustShippingLabel OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,5,8
```

--Example 08 (To print pages 3 and 8 of your big report)

```
LBLPRINT CustShippingLabel OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,8
```

--Example 09 (To print page 3 only)

```
LBLPRINT CustShippingLabel OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3
```

--Example 10 (To print pages 1 through 3 and 8 of your big report)

```
LBLPRINT CustShippingLabel OPTION PRINTER +
```

```
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 1-3,8
```

--Example 11 (To print pages 3, 5, and 9 through 15 of your big report)

```
LBLPRINT CustShippingLabel OPTION PRINTER +
```

```
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,5,9-15
```

1.13.4.3 BMP

The BMP output option displays the label in a Bitmap image file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION BMP
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether the file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.

PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print as BMP file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION BMP +
|FILENAME CusrLab128.BMP +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.13.4.4 DIF

The DIF output option displays the label in a Data Interchange Format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION DIF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.

SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print label as DIF file
PRINT CustomerLabels WHERE CustID = 128 +
OPTION DIF +
|FILENAME CustLab128.DIF +
|SHOW_CANCEL_DIALOG OFF +
```

```
| OPEN OFF +
| ROW_SPACING 1
```

1.13.4.5 EMF

The EMF output option displays the label in an Enhanced Metafile image file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION EMF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a

		band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as EMF file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION EMF +
|FILENAME CustLabl28.EMF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.13.4.6 ETXT

The ETXT output option displays the label in an emulation text file, which prints the label as plain readable text including all bands, such as Column Header, Break Headers, Break Footers, and Column Footers, if exist.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION ETXT
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
APPEND		Appends the report data to the specified FILENAME value.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST	Determines the pages in which the Page Style will appear on.

	LAST Page List e.g. 1,4-6,10-12,20	
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as an Emulation Text - ETEXT file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION ETEXT +
|FILENAME CustLab128.TXT +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON
```

1.13.4.7 GIF

The GIF output option displays the label in a Graphics Interchange Format image file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION GIF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT	Specifies the way background image has to be displayed in the GIF file.

	CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print customer labels as a GIF file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION GIF +
|FILENAME CustLabl28.GIF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.13.4.8 HTML

The HTML output option displays the label in a hypertext markup language file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION HTML
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the way background image has to be displayed in the file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TY PE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.
INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.

INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment.

		See additional options when EMAIL value is set to ON. EMAIL SET ON
--	--	---

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed HTML parameter options:

OPTIMIZE_FOR_IE	ON OFF	Specifies whether the entire report output as HTML document should be optimized for the MS Internet Explorer browser.
PAGE_END_LINES	ON OFF	Indicates whether a line should be drawn at the end of each page. Setting PAGE_END_LINES to ON draws a line after each page. PAGE_END_LINES is considered only when SEP_FILE_PER_PAGE is set to OFF.
OUTPUT_STYLES_TO_CSS_FILE	ON OFF	Specifies whether to output styles to a CSS (Cascade Style Sheet) file.
CSSFILE	value	Specifies the CSS file name if OUTPUT_STYLES_TO_CSS_FILE is set to ON.
OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE	ON OFF	Specifies whether to output scripts to a Java Script File.
JAVA_SCRIPT_FILE_NAME	value	Specifies the file name if the OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE is set to ON.
GENERATE_TOC	ON OFF	Specifies whether to generate a Table of Contents based on all report Break Header(s), if defined.
TOC_FILE_NAME	value	Specifies the file name for the Table of Contents, if the GENERATE_TOC option is set to ON.
IMAGE_FOLDER	value	specify the folder path and name for HTML document images.
DEFAULT_FONT_NAME	value	Specifies the default font name for HTML document.
DEFAULT_FONT_SIZE	value	Specifies the default font size for HTML document.
DEFAULT_FONT_COLOR	value	Specifies the default font color for HTML document.
DEFAULT_FONT_BOLD	ON OFF	Specifies whether the font is bold faced
DEFAULT_FONT_ITALIC	ON OFF	Specifies whether the font is italicized.
DEFAULT_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined.
DEFAULT_FONT_STRIKEOUT	ON OFF	Specifies whether the font is displayed with a horizontal line through it.
SEP_FILE_PER_PAGE	ON OFF	Specifies whether to create a separate file for each page of the report.
NAV_SHOW	ON OFF	Set to ON to show a collection of links that enable easy navigation between pages of the exported HTML. The links function much like the navigator buttons in the standard preview of Report Preview: one link to jump to the first page; one to the last page; one to the next page and another to get back to the previous page. NAV_SHOW is only effective when the option for SEP_FILE_PER_PAGE is set to ON.
LINK_BACK_COLOR	value	Specifies the background color of navigator. This option will only be used if

		the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_BACK_COLOR	value	Specifies the background color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_FORE_COLOR	value	Specifies the color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
USE_TEXT_LINKS	ON OFF	Specifies the type of the navigator links that should be encoded for a multi-file HTML output. Setting USE_TEXT_LINKS to ON uses the text provided in LINK_TEXT_FIRST, LINK_TEXT_PREV, LINK_TEXT_NEXT and LINK_TEXT_LAST as the hyperlink's caption. Setting to OFF uses Images specified by LINK_IMG_FILE_FIRST, LINK_IMG_FILE_PREV, LINK_IMG_FILE_NEXT, LINK_IMG_FILE_LAST. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
LINK_TEXT_FIRST	value	Specifies the hyperlink's text caption for first page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_PREV	value	Specifies the hyperlink's text caption for previous page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_NEXT	value	Specifies the hyperlink's text caption for next page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_LAST	value	Specifies the hyperlink's text caption for last page when USE_TEXT_LINKS option is set to ON.
LINK_FONT_NAME	value	Specifies the font name for navigator links.
LINK_FONT_SIZE	value	Specifies the font size for navigator links.
LINK_FONT_COLOR	value	Specifies the font color for navigator links.
LINK_FONT_BOLD	ON OFF	Specifies whether the font is bold faced for navigator links.
LINK_FONT_ITALIC	ON OFF	Specifies whether the font is italicized for navigator links.
LINK_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined for navigator links.
LINK_FONT_STRIKEOUT	ON OFF	Specifies whether the font for navigator link is displayed with a horizontal line through it.
LINK_IMG_FILE_FIRST	value	Specifies the hyperlink's image file for first page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_PREV	value	Specifies the hyperlink's image file for previous page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_NEXT	value	Specifies the hyperlink's image file for next page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_LAST	value	Specifies the hyperlink's image file for last page when USE_TEXT_LINKS option is set to OFF.

NAV_TYPE	PAGE SCREEN	Specifies the type of the navigator. If NAV_TYPE is set to SCREEN then the navigator will be fixed to screen, else the navigator will be fixed to page. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_ORIENT	VERTICAL HORIZONTAL	Specifies the orientation of the navigator. if NAV_ORIENT is set to HORIZONTAL then, a Horizontal navigator is encoded, else a Vertical navigator is encoded. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_POS	TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the position of the navigator. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
DISABLE_RIGHT_CLICK	ON OFF	Specifies whether to disable the typical speed menu options when you right click on your mouse button.
DISABLE_COPYING	ON OFF	Specifies whether to disable the option to highlight and copy the text.
AUTO_MAXIMIZE_WINDOW	ON OFF	Specifies whether to automatically maximize the window when HTML document is opened.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a HTML file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION HTML +
|FILENAME CustLab128.htm +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
|INCLUDE_RICHTEXT OFF +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|IMAGE_DPI -1 +
|PIXELFORMAT 24 +
|INCLUDE_HYPERLINKS ON +
|TITLE Customer Label +
|OPTIMIZE_FOR_IE ON +
|PAGE_END_LINES ON +
|OUTPUT_STYLES_TO_CSS_FILE ON +
|CSSFILE +
```

```

|OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE ON +
|JAVA_SCRIPT_FILE_NAME +
|GENERATE_TOC ON +
|TOC_FILE_NAME Index.htm +
|IMAGE_FOLDER Images +
|DEFAULT_FONT_NAME Verdana +
|DEFAULT_FONT_SIZE 9 +
|DEFAULT_FONT_COLOR BLACK +
|DEFAULT_FONT_BOLD OFF +
|DEFAULT_FONT_ITALIC OFF +
|DEFAULT_FONT_UNDERLINE OFF +
|DEFAULT_FONT_STRIKEOUT OFF +
|SEP_FILE_PER_PAGE ON +
|NAV_SHOW ON +
|LINK_BACK_COLOR WHITE +
|LINK_H_BACK_COLOR BLUE +
|LINK_H_FORE_COLOR WHITE +
|USE_TEXT_LINKS OFF +
|LINK_TEXT_FIRST First +
|LINK_TEXT_PREV Previous +
|LINK_TEXT_NEXT Next +
|LINK_TEXT_LAST Last +
|LINK_FONT_NAME Wingdings +
|LINK_FONT_SIZE 10 +
|LINK_FONT_COLOR BLUE +
|LINK_FONT_BOLD OFF +
|LINK_FONT_ITALIC OFF +
|LINK_FONT_UNDERLINE OFF +
|LINK_FONT_STRIKEOUT OFF +
|LINK_IMG_FILE_FIRST Images\FirstOff.gif +
|LINK_IMG_FILE_PREV Images\PrevOff.gif +
|LINK_IMG_FILE_NEXT Images\NextOff.gif +
|LINK_IMG_FILE_LAST Images\LastOff.gif +
|NAV_TYPE PAGE +
|NAV_ORIENT HORIZONTAL +
|NAV_POS BOTTOMCENTER +
|DISABLE_RIGHT_CLICK OFF +
|DISABLE_COPYING OFF +
|AUTO_MAXIMIZE_WINDOW ON +
|OPEN ON

```

1.13.4.9 JPG

The JPG output option displays the label in a JPEG image file.

Syntax:

LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION JPG

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.

BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
GRAYSCALE	ON OFF	Determines whether the image output of a JPEG image is black and white, or color. Use GRAYSCALE for reading in and writing out, that is, for decompressing and compressing, a JPEG image. This property affects how the image is displayed. GRAYSCALE is used for speed in output optimization. When GRAYSCALE is set to ON, the color is separated from luminosity. Color takes the most time to decompress, so for previewing the image, GRAYSCALE can be set to True for speed. The output then contains 255 shades of gray.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
PROGRESSIVE_ENCODING	ON OFF	Determines whether an image can be progressively displayed when it is decompressed.
QUALITY	1-100	Determines the closeness of the saved, JPEG image to the original. As a side effect it also matter to the size of the saved file. Setting a larger value improves image quality but also increases file size; a smaller value decreases image quality and also reduces file size. Alter it according to the need.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are

		relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a JPG file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION JPG +
|FILENAME CustLab128.JPG +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|GRAYSCALE OFF +
|PIXELFORMAT 32 +
|QUALITY 100 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.13.4.10 LOTUS

The LOTUS output option displays the label in a Lotus 1-2-3 format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION LOTUS
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.

ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a Lotus file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION LOTUS +
|FILENAME CustLab128.WK1 +
|SHOW_CANCEL_DIALOG ON +
|OPEN OFF +
|ROW_SPACING 1
```

1.13.4.11 PDF

The PDF output option displays the label in a Portable Document Format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION PDF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the way background image has to be displayed in the PDF file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TY PE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.
INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value (0-100)	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit	Specifies the pixel format of the image file (bits/pixel).

	16 : 16bit 24 : 24bit 32 : 32bit	
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed PDF parameter options:

GENERATE_TOC	ON OFF	Specifies whether to generate a "Table of Contents" based on all report Break Headers, if defined.
USE_COMPRESSION	ON OFF	Specifies whether compression should be used.
COMPRESSION_METHOD	FASTEST NORMAL MAXCOMPRESS	Specifies the type of compression to be used to compress text and Image.

FONT_ENCODING	WIN_ANSI MAC_ROMAN MAC_EXPERT STANDART PDF_DOC	Specifies the type of font encoding to be used to encode font.
EMBED_USED_FONTS	ON OFF FULL (same as "ON" for backward compatibility) SUBSET	Specifies whether all the used fonts should be embedded in the PDF document. Set EMBED_USED_FONTS to ON to embed all the fonts used in the report. Fonts are embedded only if embedding is permitted depending on the Embedding Flags of the font.
ENCRYPT_FILE	ON OFF (default)	Specifies whether the PDF document should be encrypted.
USER_PASSWORD	value	Specifies the user password of the encrypted PDF document. Opening the document with the correct user password allows additional operations to be performed according to the user access permissions specified by the permissions properties such as, ENABLE_PRINTING (ON/OFF), ENABLE_COPYING (ON/OFF), and ENABLE_MODIFYING (ON/OFF). These permissions are considered by the viewer applications if the document is opened with the User password. If ENABLE_PRINTING is OFF then printing is not enabled. If ENABLE_COPYING is OFF then page contents cannot be selected and copied. If ENABLE_MODIFYING is OFF then the PDF document cannot be modified.
OWNER_PASSWORD	value	Specifies the owner password of the encrypted PDF document. Opening the document with the correct owner password (assuming it is not same as the user password) allows full (owner) access to the document. This unlimited access includes the ability to change the document's passwords and access permissions.
ENCRYPTION_LEVEL	40BIT 128BIT	Specifies the level of encryption of the PDF document. Set ENCRYPTION_LEVEL to 40BIT for a 40 bit encryption of the PDF document, i.e a 40 bit key is used to encrypt the PDF document. Set ENCRYPTION_LEVEL to 128BIT for a 128 bit encryption of the PDF document, i.e a 128 bit key is used to encrypt the PDF document. ENCRYPTION_LEVEL is considered only if ENCRYPT_FILE is set to ON.

ENABLE_PRINTING	ON (default) OFF	Specifies whether to allow printing when USER_PASSWORD is set.
ENABLE_COPYING	ON (default) OFF	Specifies whether to allow page contents to be selected and copied when USER_PASSWORD is set.
ENABLE_MODIFYING	ON (default) OFF	Specifies whether the PDF document can be modified when USER_PASSWORD is set.
ENABLE_ANNOTATION	ON OFF	Specifies whether the PDF document supports annotation when USER_PASSWORD is set.
ENABLE_FORM_FILL	ON OFF	Specifies whether the PDF document supports form fill when USER_PASSWORD is set.
ENABLE_ACCESSIBILITY	ON OFF	Specifies whether the PDF document supports tagging accessibility when USER_PASSWORD is set.
ENABLE_DOC_ASSEMBLY	ON OFF	Specifies whether the PDF document supports document assembly when USER_PASSWORD is set.
ENABLE_HIRES_PRINT	ON OFF	Specifies whether the PDF document supports high resolution printing when USER_PASSWORD is set.
HIDE_TOOLBAR	ON OFF (default)	Specifies whether to hide the viewer application's toolbars, such as Acrobat Reader, when the document is active. If the value is ON then the viewer applications toolbars are hidden.
HIDE_MENUBAR	ON OFF (default)	Specifies whether to hide the viewer application's menu bar, such as Acrobat Reader, when the document is active. If the value is ON then the viewer applications menu bar is hidden.
HIDE_WINDOW_UI	ON OFF (default)	Specifies whether to hide user interface elements in the document's window (such as scroll bars and navigation controls), leaving only the document's contents displayed. If the value is ON then documents user interface elements are hidden.
FIT_WINDOW	ON OFF (default)	Specifies whether to resize the document's window to fit the size of the first displayed page. If the value is ON then the first page displayed is resized to fit the document's window.
CENTER_WINDOW	ON OFF (default)	Specifies whether to position the document's window in the center of the screen. If the value is ON then the document is positioned at the center of the screen when opened.
PAGE_LAYOUT	SINGLE_PAGE ONE_COLUMN TWO_COLUMN_LEFT	Specifies the page layout to be used when the PDF document is opened. If the value is set to

	TWO_COLUMN_RIGHT	SINGLE_PAGE then one page is displayed at a time. If the value is set to ONE_COLUMN then pages are displayed in one column. If the value is set to TWO_COLUMN_LEFT then pages are displayed in two columns with odd numbered pages on the left. If the value is set to TWO_COLUMN_RIGHT then pages are displayed in two columns with odd numbered pages on the right.
PAGE_MODE	NONE OUTLINES THUMBS FULLSCREEN	Specifies how the document should be displayed when opened. If the value is set to NONE then neither document outlines nor thumbnail images are visible. If the value is set to OUTLINES then document outline is visible. If the value is set to THUMBS then thumbnail images are visible. If the value is set to FULLSCREEN then document is opened in Full Screen mode with no menu bar, window controls, or any other window visible.
NON_FULL_SCREEN_PAGE_MODE	NONE USE_OUTLINES USE_THUMB	Specifies how to display the document on exiting full screen mode. This is meaningful only if the PAGE_MODE value is set to FULLSCREEN. If the value is set to NONE then neither document outlines nor thumbnail images are visible. If the value is set to USE_OUTLINES then document outline is visible. If the value is set to USE_TNUMB then thumbnail images are visible.
PRESENTATION_MODE	SPLIT_HOR_INWARD SPLIT_HOR_OUTWARD SPLIT_VERT_INWARD SPLIT_VERT_OUTWARD BLINDS_HOR BLINDS_VERT BOX_INWARD BOX_OUTWARD WIPE_LEFT_TO_RIGHT WIPE_BOTTOM_TO_TOP WIPE_RIGHT_TO_LEFT WIPE_TOP_TO_BOTTOM DISSOLVE GLITTER_LEFT_TO_RIGHT GLITTER_TOP_TO_BOTTOM GLITTER_TOPLEFT_TO_BOTTOMRIGHT NONE	Specifies how the PDF viewer application will display/present the document with transition effects. Some viewer applications allow a document to be displayed in the form of a presentation or "slide show," advancing from one page to the next either automatically or under user control. A detailed description of each transition effect is described below.
DURATION	value	Specifies the duration of the presentation mode (transition effect) in seconds.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Transition Effects

SPLIT_HOR_INWARD	Two lines sweep across the screen, revealing the new page. The lines are horizontal and move inward from the edges of the page.
SPLIT_HOR_OUTWARD	Two lines sweep across the screen, revealing the new page. The lines are horizontal and move outward from the center of the page.
SPLIT_VERT_INWARD	Two lines sweep across the screen, revealing the new page. The lines are vertical and move inward from the edges of the page.
SPLIT_VERT_OUTWARD	Two lines sweep across the screen, revealing the new page. The lines are vertical and move outward from the center of the page.
BLINDS_HOR	Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are horizontal and move downward.
BLINDS_VERT	Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are vertical and move to the right.
BOX_INWARD	A rectangular box sweeps inward from the edges of the page revealing the new page.
BOX_OUTWARD	A rectangular box sweeps outward from the center of the page revealing the new page.
WIPE_LEFT_TO_RIGHT	A single line sweeps across the screen from Left to Right edge of the page.
WIPE_BOTTOM_TO_TOP	A single line sweeps across the screen from Bottom to the Top edge of the page.
WIPE_RIGHT_TO_LEFT	A single line sweeps across the screen from Right to the Left edge of the page.
WIPE_TOP_TO_BOTTOM	A single line sweeps across the screen from Top to Bottom edge of the page.
DISSOLVE	The old page "dissolves" gradually to reveal the new one.
GLITTER_LEFT_TO_RIGHT	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Left side of the screen to the Right side.
GLITTER_TOP_TO_BOTTOM	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Top side of the screen to the Bottom side.
GLITTER_TOPLEFT_TO_BOTTOMRIGHT	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Top Left corner of the screen to the Bottom Right corner.
NONE	The new page simply replaces the old one with no special transition effect.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a secure PDF document
LBPRIINT CustomerLabels WHERE CustID = 128 +
OPTION PDF +
|FILENAME CustLab128.PDF +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
|INCLUDE_RICHTEXT ON +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
```

```

| IMAGE_FORMAT JPG +
| PIXELFORMAT 32 +
| JPEG_QUALITY 100 +
| IMAGE_DPI -1 +
| INCLUDE_HYPERLINKS ON +
| GENERATE_TOC ON +
| TITLE Customer Label +
| USE_COMPRESSION ON +
| COMPRESSION_METHOD MAXCOMPRESS +
| FONT_ENCODING WIN_ANSI +
| EMBED_USED_FONTS ON +
| ENCRYPT_FILE ON +
| USER_PASSWORD RRBYW20 +
| OWNER_PASSWORD RRBYW2Owner +
| ENCRYPTION_LEVEL 128BIT +
| ENABLE_PRINTING OFF +
| ENABLE_COPYING OFF +
| ENABLE_MODIFYING OFF +
| HIDE_TOOLBAR OFF +
| HIDE_MENUBAR OFF +
| HIDE_WINDOW_UI OFF +
| FIT_WINDOW OFF +
| CENTER_WINDOW OFF +
| PAGE_LAYOUT SINGLE_PAGE +
| PAGE_MODE NONE +
| NON_FULL_SCREEN_PAGE_MODE NONE +
| PRESENTATION_MODE NONE +
| DURATION 0 +
| OPEN ON

```

1.13.4.12 PNG

The PNG output option displays the label in a Portable Network Graphics image file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION PNG
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report
INCLUDE_TEXT	ON OFF	Specifies whether text on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will

		increase. An example would be 200. The Default Value is - 1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
COLOR_TYPE	GRAYSCALE RGB (default) RGB_ALPHA GRAY_ALPHA	Specifies the color type of the image
PIXEL_DIMENSIONS	ON OFF	Enables the ability to set PNG pixel dimension information
UNIT_SPECIFIER	NONE METER	Specifies the units for setting the pixel dimension information. PIXEL_DIMENSIONS must be set ON in order to use the parameter. With NONE, X_PIXELS_PER_UNIT and Y_PIXELS_PER_UNIT defines pixel aspect ratio only, and the actual size of the pixels remains unspecified. Using METER the actual size of the pixels is specified
X_PIXELS_PER_UNIT	value	Specifies the number of pixels per unit in the horizontal direction. PIXEL_DIMENSIONS must be set ON in order to use the parameter. Default: 1
Y_PIXELS_PER_UNIT	value	Specifies the number of pixels per unit in the vertical direction. PIXEL_DIMENSIONS must be set ON in order to use the parameter. Default: 1
TRANSPARENCY	ON OFF	Sets the fields related to transparency of the output
IMAGE_ALPHA	0 - 100	Specifies a percentage of transparency for the output. This is applicable only when TRANSPARENCY is ON and COLOR_TYPE is either RGB_ALPHA or GRAY_ALPHA. If IMAGE_ALPHA is 0, then the image is completely transparent. If IMAGE_ALPHA is 100, then the image is completely opaque. Default: 0
TRANSPARENT_COLOR	value	Specifies the color that is transparent. This is applicable only when TRANSPARENCY is ON and COLOR_TYPE is either RGB or GRAYSCALE.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a

		band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a PNG file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION PNG +
|FILENAME CustLab128.PNG +
|SHOW_CANCEL_DIALOG OFF +
|INCLUDE_SHAPES OFF +
|INCLUDE_IMAGES ON +
|OPEN ON
```

1.13.4.13 QPRO

The QPRO output option displays the label in a Quattro Pro format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION QPRO
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for

	PAGE_LIST	PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a Quattro Pro file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION QPRO +
|FILENAME CustLab128.WB1 +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON +
|COLUMN_SPACING 1
```

1.13.4.14 RTF

The RTF output option displays the label in a rich text format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION RTF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
-----------	-------	-------------

FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
APPEND		Appends the report data to the specified FILENAME value.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
GRAPHIC_DATA_IN_BINARY	ON (default) OFF	Specifies whether graphic data is to be encoded in binary format. Set to ON to encode graphic data in binary format. If OFF, graphic data is encoded in hexadecimal format.
OPTIMIZATION	POSITIONAL_ACCURACY (default) EASE_OF_EDITING	Specifies the type of encoding of the RTF document. Resulting RTF document can be either optimized for positional accuracy (meaning that the placement of the objects will be as close as possible to a printed report), or for ease of editing (meaning that everything is treated as simple text and it is much easier to edit the file at the expense of aesthetics). Set OPTIMIZATION EASE_OF_EDITING if you wish to easily modify the resulting RTF document.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report

KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content in RTF document property summary.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a RTF document
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION RTF +
|FILENAME CustLab128.RTF +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
```

```

| INCLUDE_LINES ON +
| INCLUDE_SHAPES ON +
| INCLUDE_RICHTEXT ON +
| INCLUDE_IMAGES ON +
| IMAGE_FORMAT JPG +
| JPEG_QUALITY 100 +
| IMAGE_DPI -1 +
| PIXELFORMAT 32 +
| GRAPHIC_DATA_IN_BINARY ON +
| OPTIMIZATION POSITIONAL_ACCURACY +
| TITLE Customer Label +
| SUBJECT Sales Order +
| AUTHOR Consolidated Computer Company +
| OPEN ON

```

1.13.4.15 SVG

The SVG output option displays the label in a scalable vector graphics file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SVG
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report
INCLUDE_TEXT	ON OFF	Specifies whether text on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
IMAGE_FORMAT	GIF JPG BMP	Specifies the image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	1-100	Specifies the quality of the JPEG Image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.

PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a SVG file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SVG +
|FILENAME CustLab128.SVG +
|SHOW_CANCEL_DIALOG OFF +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|OPEN ON
```

1.13.4.16 SYLK

The SYLK output option displays the label in a Symbolik Link format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SYLK
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a Symbolik Link format file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION SYLK +
```

```

|FILENAME CustLab128.sylk +
|SHOW_CANCEL_DIALOG ON +
|OPEN ON +
|ROW_SPACING 1

```

1.13.4.17 TIFF

The TIFF output option displays the report in the tagged image file format.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION TIFF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether the file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
MULTIPAGE	ON OFF	Specifies the output report to one multi-page file or one file per page.
COMPRESSION	NONE PACKBITS GROUP3_1D GROUP3_2D GROUP4 ZLIB JPEG	When an image is loaded, this parameter indicates the compression used by the image. When saving in image, the parameter may be set to specify which compression method to use.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.

PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a TIF file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION TIFF +
|FILENAME CustLab128.TIF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|MUTIPAGE OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|COMPRESSION NONE +
|OPEN ON
```

1.13.4.18 TXT

The TXT output option displays the label in a text file format.

Caution:

This text file option requires the bands to be defined within the Label Designer. From the main Menu Bar, select "File" > "Print to Text File Setup..."

If a user attempts to print the label as a text file using the OPTION TXT parameter without first having defined the "Text File Setup" within the Label Designer, an error message will be displayed and a zero length file will be created. The same rule applies when printing a label to a text file from the Database Explorer.

For step-by-step instructions on how to set up a label to print as Text File, please see Printing Report as Text File.

Syntax:

LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION TXT

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.

- The TXT "Text File" option allows you to define a particular band to print the data as Comma Delimited, Tab Delimited or Fixed Length file. This is the perfect option to export selective and/or summarized data used in a label band.

Example:

```
--To print a label as a Text file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION TXT +
|FILENAME CustLab128.TXT +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON
```

1.13.4.19 WMF

The WMF output option displays the label in a Windows Metafile image file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION WMF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST	Determines the pages in which the Page Style will appear on.

	Page List e.g. 1,4-6,10-12,20	
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print a label as a WMF file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION WMF +
|FILENAME CustLab128.WMF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.13.4.20 XHTML

The XHTML output option displays the label in a extensible hypertext markup language file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XHTML
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT	Specifies the way background image has to be displayed in the file.

	TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TY PE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.
INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.

PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed HTML parameter options:

PAGE_END_LINES	ON OFF	Indicates whether a line should be drawn at the end of each page. Setting PAGE_END_LINES to ON draws a line after each page. PAGE_END_LINES is considered only when SEP_FILE_PER_PAGE is set to OFF.
OUTPUT_STYLES_TO_CSS_FILE	ON OFF	Specifies whether to output styles to a CSS (Cascade Style Sheet) file.
CSSFILE	value	Specifies the CSS file name if OUTPUT_STYLES_TO_CSS_FILE is set to ON.
OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE	ON OFF	Specifies whether to output scripts to a Java Script File.
JAVA_SCRIPT_FILE_NAME	value	Specifies the file name if the OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE is set to ON.
GENERATE_TOC	ON OFF	Specifies whether to generate a Table of Contents based on all report Break Header(s), if defined.
TOC_FILE_NAME	value	Specifies the file name for the Table of Contents, if the GENERATE_TOC option is set to ON.
IMAGE_FOLDER	value	specify the folder path and name for XHTML document images.
DEFAULT_FONT_NAME	value	Specifies the default font name for XHTML document.
DEFAULT_FONT_SIZE	value	Specifies the default font size for XHTML document.

DEFAULT_FONT_COLOR	value	Specifies the default font color for XHTML document.
DEFAULT_FONT_BOLD	ON OFF	Specifies whether the font is bold faced
DEFAULT_FONT_ITALIC	ON OFF	Specifies whether the font is italicized.
DEFAULT_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined.
DEFAULT_FONT_STRIKEOUT	ON OFF	Specifies whether the font is displayed with a horizontal line through it.
SEP_FILE_PER_PAGE	ON OFF	Specifies whether to create a separate file for each page of the report.
NAV_SHOW	ON OFF	Set to ON to show a collection of links that enable easy navigation between pages of the exported XHTML. The links function much like the navigator buttons in the standard preview of Report Preview: one link to jump to the first page; one to the last page; one to the next page and another to get back to the previous page. NAV_SHOW is only effective when the option for SEP_FILE_PER_PAGE is set to ON.
LINK_BACK_COLOR	value	Specifies the background color of navigator. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_BACK_COLOR	value	Specifies the background color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_FORE_COLOR	value	Specifies the color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
USE_TEXT_LINKS	ON OFF	Specifies the type of the navigator links that should be encoded for a multi-file HTML output. Setting USE_TEXT_LINKS to ON uses the text provided in LINK_TEXT_FIRST, LINK_TEXT_PREV, LINK_TEXT_NEXT and LINK_TEXT_LAST as the hyperlink's caption. Setting to OFF uses Images specified by LINK_IMG_FILE_FIRST, LINK_IMG_FILE_PREV, LINK_IMG_FILE_NEXT, LINK_IMG_FILE_LAST. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
LINK_TEXT_FIRST	value	Specifies the hyperlink's text caption for first page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_PREV	value	Specifies the hyperlink's text caption for previous page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_NEXT	value	Specifies the hyperlink's text caption for next page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_LAST	value	Specifies the hyperlink's text caption for last page when USE_TEXT_LINKS option is set to ON.

LINK_FONT_NAME	value	Specifies the font name for navigator links.
LINK_FONT_SIZE	value	Specifies the font size for navigator links.
LINK_FONT_COLOR	value	Specifies the font color for navigator links.
LINK_FONT_BOLD	ON OFF	Specifies whether the font is bold faced for navigator links.
LINK_FONT_ITALIC	ON OFF	Specifies whether the font is italicized for navigator links.
LINK_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined for navigator links.
LINK_FONT_STRIKEOUT	ON OFF	Specifies whether the font for navigator link is displayed with a horizontal line through it.
LINK_IMG_FILE_FIRST	value	Specifies the hyperlink's image file for first page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_PREV	value	Specifies the hyperlink's image file for previous page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_NEXT	value	Specifies the hyperlink's image file for next page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_LAST	value	Specifies the hyperlink's image file for last page when USE_TEXT_LINKS option is set to OFF.
NAV_TYPE	PAGE SCREEN	Specifies the type of the navigator. If NAV_TYPE is set to SCREEN then the navigator will be fixed to screen, else the navigator will be fixed to page. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_ORIENT	VERTICAL HORIZONTAL	Specifies the orientation of the navigator. if NAV_ORIENT is set to HORIZONTAL then, a Horizontal navigator is encoded, else a Vertical navigator is encoded. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_POS	TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the position of the navigator. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
DISABLE_RIGHT_CLICK	ON OFF	Specifies whether to disable the typical speed menu options when you right click on your mouse button.
DISABLE_COPYING	ON OFF	Specifies whether to disable the option to highlight and copy the text.
AUTO_MAXIMIZE_WINDOW	ON OFF	Specifies whether to automatically maximize the window when HTML document is opened.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

--To print a label as a XHTML file

```
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION XHTML +
|FILENAME CustLab128.htm +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
|INCLUDE_RICHTEXT OFF +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|IMAGE_DPI -1 +
|PIXELFORMAT 24 +
|INCLUDE_HYPERLINKS ON +
|TITLE Customer Label +
|PAGE_END_LINES ON +
|OUTPUT_STYLES_TO_CSS_FILE ON +
|CSSFILE +
|OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE ON +
|JAVA_SCRIPT_FILE_NAME +
|GENERATE_TOC ON +
|TOC_FILE_NAME Index.htm +
|IMAGE_FOLDER Images +
|DEFAULT_FONT_NAME Verdana +
|DEFAULT_FONT_SIZE 9 +
|DEFAULT_FONT_COLOR BLACK +
|DEFAULT_FONT_BOLD OFF +
|DEFAULT_FONT_ITALIC OFF +
|DEFAULT_FONT_UNDERLINE OFF +
|DEFAULT_FONT_STRIKEOUT OFF +
|SEP_FILE_PER_PAGE ON +
|NAV_SHOW ON +
|LINK_BACK_COLOR WHITE +
|LINK_H_BACK_COLOR BLUE +
|LINK_H_FORE_COLOR WHITE +
|USE_TEXT_LINKS OFF +
|LINK_TEXT_FIRST First +
|LINK_TEXT_PREV Previous +
|LINK_TEXT_NEXT Next +
|LINK_TEXT_LAST Last +
|LINK_FONT_NAME Wingdings +
|LINK_FONT_SIZE 10 +
|LINK_FONT_COLOR BLUE +
|LINK_FONT_BOLD OFF +
|LINK_FONT_ITALIC OFF +
|LINK_FONT_UNDERLINE OFF +
|LINK_FONT_STRIKEOUT OFF +
|LINK_IMG_FILE_FIRST Images\FirstOff.gif +
|LINK_IMG_FILE_PREV Images\PrevOff.gif +
|LINK_IMG_FILE_NEXT Images\NextOff.gif +
|LINK_IMG_FILE_LAST Images\LastOff.gif +
|NAV_TYPE PAGE +
|NAV_ORIENT HORIZONTAL +
|NAV_POS BOTTOMCENTER +
```



```
|DISABLE_RIGHT_CLICK OFF +
|DISABLE_COPYING OFF +
|AUTO_MAXIMIZE_WINDOW ON +
|OPEN ON
```

1.13.4.21 XLS

The XLS output option displays the label in a Microsoft Excel format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLS
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ALL_TEXT_IN_GENERAL_FORMAT	ON OFF	Specifies whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly (Integer, Double etc).
CELL_ATTR	ON OFF	Cell attributes include shading of the cell whose color is not WHITE.
INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
IMAGE_FORMAT	BMP WMF	Specifies the image file format for image controls used in the report.
INCLUDE_LINES	ON OFF (default)	Specifies whether lines on the report should be included in the exported file.
INCLUDE_RICH_TEXT	ON (default) OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TYPE	IMAGE (default) PLAINTEXT	Indicates the way in which Rich Text contents should be encoded in the output. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode RichText contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode RichText contents as unformatted text. The font used when encoding RichText as plain text is the one set in the Font property of the RichText object. When RICHTEXT_ENCODING_TYPE is set to PLAINTEXT make sure you set the Default Font property of the RichText control in report designer to true and set the required font in the Font property.
INCLUDE_SHAPES	ON OFF (default)	Specifies whether shapes on the report should be included in the exported file.
LINE_SPACE	ACTUAL 0 1 2 3 4	Specifies the spacing to be used between lines of text.

	5	
ONE_SHEET_PER_PAGE	ON OFF (default)	Specifies whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.
- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the label data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Example:

```
--To print a label as a XLS file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION XLS +
|FILENAME CustLab128.XLS +
|SHOW_CANCEL_DIALOG OFF +
```

```

|ALL_TEXT_IN_GENERAL_FORMAT OFF +
|CELL_ATTR ON +
|INCLUDE_IMAGES OFF +
|INCLUDE_LINES OFF +
|INCLUDE_RICH_TEXT ON +
|RICHTEXT_ENCODING_TYPE IMAGE +
|INCLUDE_SHAPES OFF +
|LINE_SPACE ACTUAL +
|ONE_SHEET_PER_PAGE ON +
|OPEN ON

```

1.13.4.22 XLS2003

The XLS output option displays the label in a Microsoft Excel 2003 format file.

Syntax:

```
LBLPRINT labelname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLS2003
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ALL_TEXT_IN_GENERAL_FORMAT	ON OFF	Specifies whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly (Integer, Double etc).
CELL_ATTR	ON OFF	Cell attributes include shading of the cell whose color is not WHITE.
INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
IMAGE_FORMAT	BMP WMF	Specifies the image file format for image controls used in the report.
INCLUDE_LINES	ON OFF (default)	Specifies whether lines on the report should be included in the exported file.
INCLUDE_RICH_TEXT	ON (default) OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TYPE	IMAGE (default) PLAINTEXT	Indicates the way in which Rich Text contents should be encoded in the output. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode RichText contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode RichText contents as unformatted text. The font used when encoding RichText as plain text is the one set in the Font property of the RichText object. When RICHTEXT_ENCODING_TYPE is set to PLAINTEXT make sure you set the Default Font property of the RichText control in report designer to true and set the required font in the Font property.
INCLUDE_SHAPES	ON	Specifies whether shapes on the report should be included in the exported file.

	OFF (default)	
LINE_SPACE	ACTUAL 0 1 2 3 4 5	Specifies the spacing to be used between lines of text.
ONE_SHEET_PER_PAGE	ON OFF (default)	Specifies whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.
- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the label data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Example:

```
--To print a label as a XLS file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION XLS2003 +
|FILENAME CustLab128.XLS +
|SHOW_CANCEL_DIALOG OFF +
|ALL_TEXT_IN_GENERAL_FORMAT OFF +
|CELL_ATTR ON +
|INCLUDE_IMAGES OFF +
|INCLUDE_LINES OFF +
|INCLUDE_RICH_TEXT ON +
|RICHTEXT_ENCODING_TYPE IMAGE +
|INCLUDE_SHAPES OFF +
|LINE_SPACE ACTUAL +
|ONE_SHEET_PER_PAGE ON +
|OPEN ON
```

1.13.4.23 XLSX

The XLSX output option displays the label in a Microsoft Excel Workbook format file.

Syntax:

```
LBLPRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLSX
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and filename. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ROW_SPACING	value	Specifies the spacing between spreadsheet rows.
COLUMN_SPACING	value	Specifies the spacing between spreadsheet columns.
INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
INCLUDE_TEXT	ON OFF	Specifies whether text is included in the exported file.
PAGES_PER_WORKSHEET	ON OFF	Specifies the page limit per worksheet.
PRINT_GRID_LINES	ON OFF	Specifies whether grid lines are displayed in the exported file.
PASS_SETTINGS	value	Specifies the number of times a report traverses the data before generating pages.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.

- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the report data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Example:

```
--To print a label as a XLS file
LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION XLS +
|FILENAME CustLab128.XLS +
|SHOW_CANCEL_DIALOG OFF +
|ROW_SPACING 1 +
|COLUMN_SPACING 1 +
|PRINT_GRID_LINES ON +
|OPEN ON
```

1.13.4.24 EMAIL SET ON

The following OPTION parameters are available when the **EMAIL** value is set to **ON**:

Parameter	Value	Description
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment.
EMAIL_AUTHENTICATION	ON OFF	Specifies if the SMTP Server requires user authentication.
EMAIL_HOST	value	Specifies Hostname/IP Address of SMTP Server (Outgoing). Also sometimes known as your 'Sending Mail Server', your SMTP server receives and processes all of your outgoing mail. For example, your Outgoing Mail Server might be called something like 'my-internet-provider.com' or 'outgoing-mail.mycompany.com'. The SMTP server name is sometimes the same as your incoming mail server name. If your ISP or your network administrator has not given this server name to you, you can leave the box blank. If you do not know what your outgoing mail server name is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_PORT	value	Specifies the integer value of SMTP Server Port. (Default = 25).
EMAIL_USERID	value	Specifies a valid User Name or Login Name for SMTP Server. This is the name you use to log into your email account. Your ISP or organization provides it. It is generally, but not necessarily, the name that appears before the at sign (@) in your return email address. For example, if your email address was <jim@my-server.com> - then your username would be 'jim' (no quotes). If you do not know what your username is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_PASSWORD	value	Specifies a valid SMTP Server Password for the user.
EMAIL_ATTACHMENTS	value	Specifies the attached file list. Each file must be separated by the pound character (#).
EMAIL_BODY	value	Specifies the body of the mail.
EMAIL_DATE	value	Specifies the date the message was sent. (Default = .#DATE)
EMAIL_FROM_ADDRESS	value	Specifies the address of the E-mail account from which the mail is to be sent. Sender E-Mail Address (From E-Mail Address:) Enter your e-mail address here. For example,

		<p>your email address might be 'jim@my-internet-service.com' or 'molly@my-business-domain.com'</p> <p>Take care not to make a mistake here - or else it will make it difficult for others to reply to your emails. If you do not know what your e-mail address is, you need to contact your Internet Service Provider or Network Administrator.</p>
EMAIL_FROM_NAME	value	Specifies the person's name in an E-mail address. Your real name or what you want to appear as your name on messages that you send out.
EMAIL_TO_LIST	value	Specifies the recipient(s) of a message. Specify the list of valid e-mail address(s) to which the e-mail is to be sent.
EMAIL_REPLY	value	Specifies the delivery address for message replies. This is typically the same as Sender E-Mail Address. Enter your e-mail address here. For example, your email address might be 'jim@my-internet-service.com' or 'molly@my-business-domain.com' Take care not to make a mistake here - or else it will make it difficult for others to reply to your emails. If you do not know what your e-mail address is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_SUBJECT	value	Specifies the subject for the message. Subject is a text string that identifies the subject for the message.
EMAIL_CC_LIST	value	Specifies the list of e-mail addresses as Carbon Copies to be sent along with the e-mail.
EMAIL_BCC_LIST	value	Specifies the list of e-mail addresses as Blind Carbon Copies to be sent along with the mail.
EMAIL_DELETE_AFTER_SEND	ON OFF	Specifies if the report files are deleted after sending the e-mail.
EMAIL_SHOW_DIALOG	ON OFF	Specifies if the e-mail setup dialog is displayed.

Note: Each additional OPTION parameter must be separated by pipe | symbol

Example:

```

LBLPRINT CustomerLabels WHERE CustID = 128 +
OPTION PDF|FILENAME CustLabl128.PDF +
|EMAIL ON +
|EMAIL_TO_LIST yourname@yourdomain.com, whoelse@yourdomain.com +
|EMAIL_FROM_NAME John D. Johnson +
|EMAIL_FROM_ADDRESS johndjohnson@yourdomain.com +
|EMAIL_SUBJECT Imagine The Possibilities! +
|EMAIL_BODY Attached is the invoice - sent via R:BASE. Because we Can! +
|EMAIL_ATTACHMENTS C:\Letters\PrivacyPolicy.pdf#C:\Letters\PaymentPolicy.pdf +
|EMAIL_DELETE_AFTER_SEND ON +
|EMAIL_SHOW_DIALOG OFF

```

1.13.5 LIST

Use the LIST command to display or print information about a database.

LIST	CASCADE	_____	ASC DESC	
	COLUMNS	_____		
	FORMS	_____		
		FOR tblname	_____	
		formname	_____	
	LABELS	_____		
		FOR tblname	_____	
		labelname	_____	
	REPORTS	_____		
		FOR tblname	_____	
	reportname	_____		
SYS_	%	_____		
TABLES	_____			
UKEYS	_____			
	viewname	_____		

LIST	ALL	_____
	AUTONUM	_____
	CASCADE	_____
	COLUMN	colname
		QUALKEYS
	COMPUTED	_____
	CONSTRAINTS	FOR tblname
	CURSORS	_____
	DATABASES	_____
	DEFAULT	_____
	FKEYS	FOR tblname
		FOR tblname
	FUNCTIONS	_____
	INDEXES	indexname
		FOR tblname
	PKEYS	FOR tblname
	PROCEDURE	procname
		FOR tblname
	RULES	FOR tblname
	TABLE	tblname
		dbase
		LOCK
		SERVER
	SYS_	
	TEMP	
TRIGGERS	FOR tblname	
UKEYS	FOR tblname	

Options

ALL

Displays by table the column numbers, column names, data types, sizes, indexes, autonumbering status, expressions for computed columns, default values, column descriptions, table description, and row count for each table in the open database. A blank line will be inserted between each table.

ASC

Specifies to sort the information output alphabetically in ascending order.

AUTONUM

Displays the column name, table name, and attributes for each autonumbered column in the database.

CASCADE

Displays a list of tables with the CASCADE flag, and the type of flag; UPDATE, DELETE, or BOTH.

COLUMNS

Displays the name, data type, size, table name, index status, autonumbering status, expression, any default value, and description for each column in the open database.

COLUMN colname

Displays the name, data type, size, table name, index status, autonumbering status, expression, any default value, and description for the column that you enter in every table. If you specify part of a column name followed by a wildcard character, R:BASE will only show the column names that match that pattern.

COLUMN QUALKEYS

Displays the column name, table name, and attributes for each QualKey column in the database.

COMPUTED

Displays the column name, table name, and attributes for each computed column in the database.

CONSTRAINTS

Displays the name, type, table name, and referenced table for each constraint in the database.

CONSTRAINTS FOR tblname

Specifies a table name for which to list constraints.

CURSORS

Specifies the status of cursors and cursor definitions.

DATABASES

Lists all databases in the current directory, including the date/time stamp when the database was last opened, the last data modification date/time stamp, and the last schema modification date/time stamp.

dBASE

Lists all dBASE tables attached to the database.

DEFAULT

Displays the column name, table name, and attributes for each column with a default value definition in the database.

DES

Specifies to sort the information output alphabetically in descending order.

FKEYS

Displays all defined foreign keys detailing the key ID number, type of key (foreign), table name the key is located in, column name of the foreign key in the table, table name the foreign key references, and the column name the foreign key references.

FKEYS FOR tblname

Displays all defined foreign keys for a table detailing the key ID number, type of key (foreign), table name the key is located in, column name of the foreign key in the specified table, table name the foreign key references, and the column name that the foreign key references.

FORMS

Displays the name, driving table or view, description, date/time stamp, and version for every form in the open database.

FORMS formname

Displays the name, driving table or view, description, date/time stamp, and version for *formname*. If you specify part of a form name followed by a wildcard character, R:BASE will only show the form names that match that pattern.

FORMS FOR tblname

Displays the name, driving table or view, description, date/time stamp, and version for all forms defined for *tblname*.

FUNCTIONS

Displays all defined DLL functions.

INDEXES

Displays all defined indexes detailing the name, table name, column name(s), and if the entry is a [Not NULL](#) index.

INDEXES FOR tblname

Displays all defined indexes for a table detailing the name, table name, if the entry is a Not NULL index, column name(s), data types, orders, and sizes.

INDEXES indexname

Displays the name, table name, if the entry is a [Not NULL](#) index, column name(s), data types, orders, and sizes for an index name.

LABELS

Displays the name, driving table or view, description, date/time stamp, and version for every label in the open database.

LABELS labelname

Displays the name, driving table or view, description, date/time stamp, and version for *labelname*. If you specify part of a label name followed by a wildcard character, R:BASE will only show the label names that match that pattern.

LABELS FOR tblname

Displays the name, driving table or view, description, date/time stamp, and version for all labels defined for *tblname*.

LOCK

Lists all locked tables within the database.

PKEYS

Displays all defined primary keys detailing the key ID number, the type of key (primary), if it is referenced in another table, and the table name the key is located in.

PKEYS FOR tblname

Displays all defined primary keys for a table detailing the key ID number, type of key (primary), column name of the primary key in the specified table, table name that references the primary key, and the column name that references the primary key.

PROCEDURE

Displays the name and description for every procedure in the open database.

PROCEDURE procname

Displays the name, description, ID, date last modified, version, locked by (if locked) and return type for the specified procedure. Also, if the procedure was stored with parameters, the number of parameters, names and descriptions will be displayed.

REPORTS

Displays the name, driving table or view, description, date/time stamp, and version for every report in the open database.

REPORTS reportname

Displays the name, driving table or view, description, date/time stamp, and version for *reportname*. If you specify part of a report name followed by a wildcard character, R:BASE will only show the report names that match that pattern.

REPORTS FOR tblname

Displays the name, driving table or view, description, date/time stamp, and version for all reports defined for *tblname*.

RULES

Displays whether rules checking is set on or off, the table name, whether the conditions in the WHERE clause must be met, the rule message, and the WHERE clause for every rule in the open database.

RULES FOR tblname

Displays whether rules checking is set on or off, the table name, whether the conditions in the WHERE clause must be met, the rule message, and the WHERE clause for all rules defined for *tblname*.

SERVER

Lists all Server tables attached to the database.

SYS_%

Displays a summary of system tables and views.

TABLES

The LIST TABLES command shows the database [comment](#), number of tables, columns, indexes, views, stored procedures, forms, reports, and labels in the database, followed by the number of users connected, the last data modification date/time stamp, and the last schema modification date/time stamp. Then, the LIST command displays the database name, table name, number of columns, number of rows for each table in the open database, and lists all tables, attached dBASE files, and attached foreign server tables. These numbers all include system tables.

TABLES SYS_%

Displays a detailed list of system tables and views.

TABLES tblname

Displays the name and description of the table, table number, name, data type, size, index status, autonumbering status, expression, default value, description, constraints, and index types for each column in the table. Columns are displayed in the order they were defined. The same output is displayed if you enter LIST *tblname* as a command.

TEMP

Lists all temporary tables for the database.

TRIGGERS

Displays all of the tables that have triggers and their triggers in the open database.

TRIGGERS FOR tblname

Displays the triggers for the specified table.

UKEYS

Displays the key ID number, the type of key (unique), and the table name the key is located in.

UKEYS FOR tblname

Displays the key ID number, the type of key (unique), and the column name the unique key is assigned to.

VIEWS

Displays the database name and the names of all the views in the open database. The list displays the view name, number of columns, comment in the open database, along with the table(s)/view(s) that the view is based upon. Temporary views will display a "(T)" in front of the view name.

VIEWS viewname

Displays the name and the [SELECT](#) clause defined for the view. If you specify part of a view name followed by a wildcard character, R:BASE will only show the name and the [SELECT](#) clause that match that pattern.

About the LIST Command

If the LIST command is entered without any options, R:BASE executes the LIST TABLES command. The total number of tables and columns listed includes R:BASE system tables (of which some are hidden) and their associated columns.

You can use wildcards to display objects that match a specific pattern.

Column numbers (#1, #2, etc.) that are displayed with the LIST command can be entered in other commands instead of column names. The following command would display the first column, *custid*, in the *customer* table.

```
SELECT #1 FROM customer
```

Table Locks

LIST displays locked tables with a letter next to the table name for the type of multi-user lock.

- (L) for a local lock
- (R) for a remote lock
- (C) for a cursor lock
- (r) for a row lock

Temporary Tables

When a table is a TEMPORARY table, the LIST command will display "(T)" in front of the table name. When a table is an INMEMORY temporary table, the LIST command will display "(I)" in front of the table name.

Database Access Rights with LIST

Before using the LIST command, enter the correct user identifier with the [CONNECT](#) or [SET USER](#) command.

When access rights have been assigned to a user with the GRANT command, R:BASE displays only those tables, columns, and views for which the user has been granted access.

In a multi-user environment, locked tables are displayed in reverse video when the LIST command is run. When you run the LIST TABLE *tblname* command, R:BASE tells you if the lock is an edit, cursor, local, or remote lock.

Examples

Example 01.

-- The following lists tables in a database

```
LIST TABLES
```

```
Number of Tables in Database RRBYW20 is 81.
Number of Columns in Database RRBYW20 is 355.
Number of Indexes in Database RRBYW20 is 57.
Number of users connected: 1.
Last database modification: 12/06/2022 04:09 PM
Last schema modification: 12/06/2022 03:03 PM
```

Tables in the Database RRBYW20

Name	Columns	Rows	Comments
-----	-----	-----	-----
BonusRate	3	14	Rates for Bonuses
Component	2	12	Component Identification Number and Description
CompUsed	2	22	Components Used in a Model
Contact	12	36	Customer Contact Information
ContactCallNotes	5	3	Contact Call Notes
Customer	17	30	Customer Information
Departments	8	31	Departments
Employee	18	12	Employee Information
FormTable	1	1	Dummy Table for Forms
HourlyTemps	3	3	Temperature Data for Gauge
InvoiceDetail	8	148	Invoice Detail Information
InvoiceHeader	18	97	Invoice Header Information
Levels	2	40	Pricing Level Per Model by Percent

LicenseInformation	19	1	Running R:BASE Your Way (Part 20) - Customer License
PaymentTerms	1	8	Payment Terms
PrintOptions	6	13	Print Options
ProdLocation	5	20	Product Location, Quantity on Hand, and Cost
Product	4	8	Model Information, Including List Price
RThemes	2	86	List of New Available R:Themes
SalesBonus	5	13	Sales Bonus Information
SecurityTable	4	3	Application MainMenu Security/Login Table
Staff	8	42	Staff
StateAbr	2	51	State Names and Abbreviations
Tasks	12	28	List of Tasks to Track
Titles	2	8	Employee Titles List

Example 02.

-- The following lists views in a database

R>LIST VIEWS

Views in the Database RRBYW20

Name	Columns	Rows	Comments
CustomerContact	8	N/A	List of Customers and Contacts
CustomerList	2	N/A	View - Customer List
CustomerView	7	N/A	Sorted List of Customers
FullOuterJoin	34	N/A	Sample Full Outer Join View
InvoicesMaster	32	N/A	Invoice Master
LeftOuterJoin	2	N/A	Sample Left Outer Join View
ListOfReportsView	2	N/A	List of Reports to be Shown in Combo/List Boxes
NewContact	3	N/A	Quick List of Customer Contacts
NewCustomer	7	N/A	Test View for Sub-Reports
NewEmployee	9	N/A	Test View for Sub-Reports
NewInvoiceDetails	11	N/A	Test View for Sub-Reports
NewInvoiceHeader	14	N/A	Test View for Sub-Reports
OrganizationView	6	N/A	Organization View
ProdAlias	31	N/A	Products Sold by Date - View
ProductInvoice	21	N/A	Product Invoice View
ProdView	29	N/A	Sales by Products - View
QrtrlySmmry_ByCust	2	N/A	Quarterly Summary View (By Customer)
QuarterlySummary	2	N/A	Quarterly Summary View
SalesByEmployee	6	N/A	Products Sold by Employee - View
SalesCalendarView	11	N/A	Sales Calendar View
SalesData	9	N/A	Sales Summary by Invoice Number
SalesDataByCompany	9	N/A	Sales Data by Company
SalesInf	4	N/A	Products Sold by Employee - View
StaffByDept	11	N/A	Staff by Department
TCrossTabView	4	N/A	CrossTab Data - View
UnionView	2	N/A	Sample Union View
YT3	6	N/A	YTD Invoice Summary
YTDInvoiceTotal	6	N/A	Year-To-Date Invoice Totals by Customer

Example 03.

-- The following lists foreign keys in a database

R>LIST FKEYS

Id	Type	Table Name	Column Name(s)	Ref Table Name	Ref
Column Name(s)					
#46	FOREIGN KEY	CompUsed	CompID	Component	
CompID					
#45	FOREIGN KEY	CompUsed	Model	Product	Model
#47	FOREIGN KEY	ProdLocation	Model	Product	Model
#49	FOREIGN KEY	InvoiceDetail	Model	Product	Model
#48	FOREIGN KEY	InvoiceDetail	TransID	InvoiceHeader	
TransID					
#51	FOREIGN KEY	ContactCallNotes	EmpID	Employee	EmpID
#50	FOREIGN KEY	ContactCallNotes	ContID	Contact	
ContID					
#57	FOREIGN KEY	Levels	Model	Product	Model
#39	FOREIGN KEY	Employee	EmpTID	Titles	
EmpTID					
#54	FOREIGN KEY	Contact	CustID	Customer	
CustID					
#55	FOREIGN KEY	Staff	DepartmentID	Departments	
DepartmentID					

Example 04.

-- The following lists computed columns in a database

R>LIST COMPUTED

Computed Columns

Name

Table

Attributes

Bonus	SalesBonus	Type: CURRENCY Expression: (netamount*bonuspct)
Freight	InvoiceHeader	Type: CURRENCY Expression: (netamount* .01)
Tax	InvoiceHeader	Type: CURRENCY Expression: (netamount* .081)
InvoiceTotal	InvoiceHeader	Type: CURRENCY Expression: (netamount+freight+tax)
SalePrice	InvoiceDetail	Type: CURRENCY Expression: (Price- (Price*Discount/100))
ExtPrice	InvoiceDetail	Type: CURRENCY Expression: (Units*SalePrice)

Example 05.

-- The following lists forms that begin with "search"

R>LIST FORMS SEARCH%

Form	Table / View	Form Description
Modified	Version	
SearchAndEditCust1		Search and Edit Customers (Part 1) - Tr
01/16/2020 04:30 PM	2	

SearchAndEditCust2		Search and Edit Customers (Part 2) - On
01/16/2020 04:30 PM	2	
SearchAndPrintInv	FormTable	Search and Print Customer Invoice
10/06/2018 09:24 PM	1	
SearchCustInvoice	FormTable	Search Customer Invoice and the option t
10/06/2018 09:24 PM	1	
SearchForm_DBGGrid	Customer	Search Form using Custom DB Grid
10/06/2018 09:24 PM	1	

Example 06.

-- The following lists reports that begin with "cust" in descending order

R>LIST REPORT CUST% DES

Report Modified	Table / View Version	Report Description
-----	-----	-----
CustYellowPages	Customer	Customer Phone Directory (Yellow Pages)
10/06/2018 09:24 PM	1	
CustomPageInvoice	ProdAlias	Invoice with Custom Page Numbers
09/19/2022 11:26 AM	5	
Customers	Customer	Master List of Customers
09/20/2022 10:01 AM	5	
CustomerList	Customer	List of Active Customers
09/30/2022 10:06 AM	33	
CustomerDirectory	Customer	Customer Phone Directory
10/06/2018 09:24 PM	1	
CustomerContacts	Customer	Sub-Reports (2 Tables): Customer and Con
10/06/2018 09:24 PM	1	
CustDirWithURL_VAR	Customer	Customer Phone Directory with Hyperlinks
10/06/2018 09:24 PM	1	
CustDirWithURL_DB	Customer	Customer Phone Directory with Hyperlinks
10/06/2018 09:24 PM	1	
CustContInvoices	NewCustomer	Sub-Reports (4 Tables): NewCustomer, New
10/06/2018 09:24 PM	1	

1.13.6 LIST ACCESS

Use the LIST ACCESS command to allow the owner of a database to view all user's access rights assigned by the [GRANT](#) command, and to allow users to view their access rights and rights granted to PUBLIC.

LIST ACCESS	ALL _____ ALTER _____ CREATE _____ DELETE _____ INSERT _____ SELECT _____ UPDATE _____ REFERENCES _____	FOR user] [ON tblview]
-------------	--	---------------------------

Options

ALL

Displays the users and tables to which the ALL PRIVILEGES access right has been assigned.

ALTER

Displays the users and tables to which the ALTER access right has been assigned.

CREATE

Displays the users and tables to which the CREATE access right has been assigned.

DELETE

Displays the users and tables to which the DELETE access right has been assigned.

FOR user

Displays the tables and access rights for a specific user.

INSERT

Displays the users and tables to which the INSERT access right has been assigned.

ON tblview

Displays the users and their access rights for a specific table or view.

REFERENCES

Displays the users who have rights to reference a primary-key or unique-key table.

SELECT

Displays the users and tables to which the SELECT access right has been assigned.

UPDATE

Displays the users, tables, and columns to which the UPDATE access right has been assigned.

About the LIST ACCESS Command

The LIST ACCESS command is available when access rights have been granted. When access rights have been granted with the WITH GRANT OPTION, the LIST ACCESS command displays an asterisk in front of the access right. When the LIST ACCESS command is entered without any options, it displays all access rights that have been granted on all tables to which the current user identifier has access.

The LIST ACCESS command requires a user identifier-either the database owner's, user's, or NONE. When you enter the owner's user identifier, the LIST ACCESS command allows you to view all current access rights assigned with the [GRANT](#) command. When you enter your user identifier, the LIST ACCESS command displays only the access rights granted to PUBLIC and to you. Users other than the database owner who are assigned access rights with the WITH GRANT OPTION are able to display the access rights they grant to other users.

Examples

The following commands assign access rights.

```
GRANT INSERT ON employee TO mike, dale
GRANT SELECT, INSERT ON transmaster TO dawn +
    WITH GRANT OPTION
GRANT UPDATE ON transmaster TO andrea
GRANT UPDATE (company, custaddress, custcity) +
    ON customer TO mike, PUBLIC
```

Based on the access rights assigned with the GRANT commands above, the LIST ACCESS command would display the information below. Column names listed under Column are columns that are specified with the UPDATE access right. Columns are only displayed when a user has been assigned UPDATE rights to specific columns.

User Identifier	Table	Access	Column
MIKE	Employee	INSERT	
DALE	Employee	INSERT	
DAWN	TransMaster	*SELECT	
DAWN	TransMaster	*INSERT	
ANDREA	TransMaster	UPDATE	
MIKE	Customer	UPDATE	company custaddress custcity
PUBLIC	Customer	UPDATE	company custaddress custcity

Based on the access rights assigned with the GRANT commands above, the LIST ACCESS UPDATE command would display the information below.

User Identifier	Table	Access	Column
ANDREA	TransMaster	UPDATE	
MIKE	Customer	UPDATE	company custaddress custcity
PUBLIC	Customer	UPDATE	company custaddress custcity

Based on the access rights assigned with the GRANT commands above, the LIST ACCESS ON *transmaster* command would display the information below. The asterisk (*) by SELECT and INSERT shows that Dawn can grant those access rights to other users.

User Identifier	Table	Access	Column
DAWN	TransMaster	*SELECT	
DAWN	TransMaster	*INSERT	
ANDREA	TransMaster	UPDATE	

1.13.7 LOAD

Use the LOAD command to add data to a table or to a single table view that can be updated.

```
LOAD tblview WITH PROMPTS [ FOR n ROWS ][ USING collist ][ REINDEX ]

LOAD tblview FROM filespec [ AS FORMATTED USING colname start end ][ REINDEX ]
                           [ AS ASCII ][ AS CSU ]

LOAD tblview FROM filespec [ FOR n ROWS ][ USING collist ] ...

... [ ; CHECK ][ ; FILL ][ ; NUM ][ REINDEX ]
    [ ; NOCHECK ][ ; NOFILL ][ ; NONUM ]

LOAD tblview [ FOR n ROWS ][ USING collist ][ REINDEX ]
CHECK/NOCHECK
FILL/NOFILL
NUM/NONUM
data=block
END
```

Options

Indicates that this part of the command is repeatable.

AS ASCII

The AS ASCII command parameter is designed strictly for speed of operation. AS ASCII checks rules and constraints. However, following is a list of the limitations of the AS ASCII command parameter:

- It does not check the data types; therefore, invalid data will be loaded as null values into columns; no error messages about this conversion are displayed.
- It does not display error messages when columns must be truncated, or when excess data exists on any line.
- It does not echo data to the screen regardless of the setting for [SET ECHO](#).

To achieve maximum speed of loading, the data must look like the data that R:BASE unloads with the UNLOAD DATA AS ASCII command. That is, the data must conform to the following:

- The carriage return/line feed characters define the end of the line for a given row.
- The maximum row size is 32,768 characters.
- The data cannot include variables.
- The data cannot include comments.

AS FORMATTED USING

Loads data from an ASCII file when the data is formatted in fixed column locations, with the following restrictions:

- The carriage return/line feed characters define the end of the line for a given row.
- The maximum row size is 32,768 characters.
- The data cannot include variables.
- The data cannot include comments.
- You must specify the name of each column of the table to be loaded, and the starting and ending position of its data in the line, which is specified in the USING clause of this command.

AS CSV

Loads data from a comma-separated values (CSV) file. AS CSV checks rules and constraints. The following is a list of the limitations of the AS CSV command parameter:

- It does not check the data types; therefore, invalid data will be loaded as null values into columns; no error messages about this conversion are displayed.
- It does not display error messages when columns must be truncated, or when excess data exists on any line.
- It does not echo data to the screen regardless of the setting for [SET ECHO](#).

The data must look like the data that R:BASE unloads with the UNLOAD DATA AS CSV command. That is, the data must conform to the following:

- The carriage return/line feed characters define the end of the line for a given row.
- The maximum row size is 32,768 characters.
- The data cannot include variables.
- The data cannot include comments.

CHECK

NOCHECK

CHECK turns on rule checking. When rule checking is on, R:BASE checks input against data validation rules. NOCHECK turns off rule checking. CHECK and NOCHECK override the current setting of the [SET RULES](#) condition. The default is CHECK.

colname start end

Specifies the name of a column in the table and the starting and ending position of its data in the line; this option is used with the AS FORMATTED option.

data-block

Includes lines of data to be loaded, as well as the LOAD subcommands.

For these data types...	Use this format for <i>data-block</i>
All data types except BIT, BITNOTE, LONG VARBIT, and VARBIT	'string' or value

BIT, BITNOTE, LONG VARBIT, LONG VARCHAR, VARBIT, and VARCHAR

['filename.ext'] or ['filename.ext', filetype, offset, length]

Note: When you use VARCHAR, the filetype is always TXT. When you use VARBIT, BIT, and BITNOTE, filetype refers to the standard graphical file types.

FILL NOFILL

FILL makes null any columns that have not been assigned values. All of the missing values must be at the end of the row. If a rule specifies that a column requires an entry other than null, do not use FILL. NOFILL turns off FILL and requires a value for each column. The default is NOFILL.

FOR n ROWS

Directs R:BASE to stop processing after loading *n* rows, where *n* is a positive whole number. In the fourth syntax diagram, END is not used if FOR *n* ROWS is included.

FROM filespec

Loads data into the specified table with data from an external file.

NUM NONUM

NUM specifies that autonumbering columns will be numbered as they are loaded. NONUM turns off autonumbering while loading, thereby allowing loading of a specific value for autonumber columns. The default is NUM.

REINDEX

Specified that R:BASE will temporarily add the new rows without updating every non-constraint index row by row, and only updates constraint indexes row by row. When the new rows are done being added, the non-constraint indexes are then rebuilt. The rebuilding of the non-constraint indexes after the rows are added increases the overall speed of the operation.

tblview

Specifies a table or view name to load.

USING collist

Specifies the column(s) to use with the command.

WITH PROMPTS

Loads data into the specified table from keyboard entries. R:BASE asks for the values of each column by displaying the column name and its data type. To end the loading session, press [Esc].

About the LOAD Command

You cannot load data into a multi-table view.

Instead of using LOAD, you can also use [INSERT](#), the Data Editor, or a Form to add data to a table.

You can use the LOAD command to load data into R:BASE from a file that was not created by R:BASE. The file must be an ASCII file (delimited or fixed), or a CSV file.

The LOAD command will differentiate between END and 'END'; FILL and 'FILL'; NOFILL and 'NOFILL'; CHECK and 'CHECK'; NOCHECK and 'NOCHECK'; NUM and 'NUM'; NONUM and 'NONUM'. So, make sure to use the proper syntax when creating LOAD statements.

To ensure that data is placed in the intended column, use the following guidelines:

- Do not embed commas within entries for CURRENCY, DATE, DATETIME, DOUBLE, INTEGER, NUMERIC, or REAL data types. R:BASE automatically inserts commas and the current currency symbol.
- When values for CURRENCY, DOUBLE, NUMERIC, or REAL or data types are decimal fractions, you must enter the decimal point. When values are whole numbers, R:BASE adds a decimal point for you at the end of the number. R:BASE adds zeros for subunits in whole currency values. For example, using the default currency format, R:BASE loads an entry of 1000 as \$1,000.00.
- When values for NOTE or TEXT data types contain commas, you can either enclose the entries within quotes, or use [SET DELIMIT](#) to change the default delimiter (comma) to another character.

- When values for NOTE or TEXT data types contain single quotes ('), and you are using the default QUOTES character ('), use two single quotes (') in the text string. For example, 'Walter Finnegan's order.'
- When a value you specify for a column is not the same data type as the column's data type, R:BASE displays an error message and you need to re-enter the entire row.
- When values for NOTE or TEXT data types exceed the maximum length of a column, R:BASE truncates the value and adds it to the table. A message is displayed that tells you which row has been truncated.

Loading with a USING Clause

A USING clause is helpful when you do not have all the information that is to be added to a table. The following example lets you enter some information for a product but does not require that all columns be entered. The *model* and *listprice* columns are the first and last columns in the *product* table. The *prodname* and *proddesc* columns are not included in the command and are loaded with null values. You can later edit the *product* table to enter data into the columns that have null values.

```
LOAD product USING model listprice
```

Loading with the CHECK Option

The [SET RULES](#) condition does not have any effect on the CHECK option because CHECK has precedence over a RULES setting. When RULES is set off, the CHECK option still verifies data entry against existing rules.

When a user identifier has been assigned to the database owner, you must enter the owner's user identifier with the [CONNECT](#) or [SET USER](#) command before you use the CHECK or NOCHECK option. R:BASE does not accept the CHECK or NOCHECK option unless the owner's user identifier has been entered.

Loading Computed Columns

By default, you cannot load data directly into a computed column. After you load the column values that are used to calculate the computed column, R:BASE fills the computed column with the computed value. To load data into a computed column, the [NOALCALC](#) setting must be set ON.

The setting of the [SET ZERO](#) condition affects the calculation of numeric computed columns. To have null values treated as zeros in expressions, set ZERO on. When ZERO is set off, if the value of a column used in an expression is null, the computed value will be null.

Loading Negative CURRENCY values

When loading negative CURRENCY values into a table, the format must include the hyphen, i.e. -\$500.00. Negative CURRENCY values encased in parenthesis are not recognized, e.g. (\$500.00).

Loading with Prompts

When you run the LOAD command using prompts, you load one row of data at a time into the table you specified. (See Example 1). For each new row you add, R:BASE displays the name and data type of the row's column as prompts. At each prompt, you enter the value that you want the column to contain. You are prompted for each column in the row beginning with the first column, unless you used a USING *collist* clause to limit the number of the columns to load, or to change the order in which the columns are loaded. Any columns not listed in the *collist* are given null values when the rows are entered.

When you load data with prompts, the default length for a text entry is 80 characters. To enter columns with a NOTE or TEXT data types that contain more than 80 characters, load the data without prompts, make a custom data-entry form, or set the WIDTH so you can enter more characters.

R:BASE does not prompt you for computed or autonumbered column values.

Loading without Prompts

Loading without prompts is faster but requires that you remember the order of the columns in the table. When you load without using prompts and not from a file, the LOAD command provides its own distinctive prompt. The following options can be entered at this prompt: CHECK, NOCHECK, FILL, NOFILL, NUM, and NONUM.

Loading from an ASCII/CSV File

Use the LOAD command to load data into an existing table from both delimited and fixed field ASCII files, as well as CSV files. Each record in the file corresponds to one row of data in a table, and each item of

data in a record corresponds to one column value in a row. Therefore, organize data in the file in the same order as the columns in the table to be loaded.

Items of data in a line of the ASCII/CSV file must be delimited to be properly placed within the columns of a row. The delimiter character must be the same as the current delimiter character specified with the [SET DELIMIT](#) setting. (The default delimiter is a comma.) R:BASE also accepts a blank space as a delimiter, regardless of the setting of the DELIMIT setting.

Data can be loaded in a fixed-field formatted ASCII file with the AS FORMATTED option. The column name and the start and end positions within the file must be specified for each value in the row of data that is to be loaded. When the start and end positions are specified, the delimiter character does not have any effect because the start and end positions for each column identify the data.

When loading from a file, be sure that the current null symbol is not a blank. If the first four characters of a field in a file are blank, R:BASE adds the column as a null column and does not read any additional characters that make up the field value.

When loading data from an ASCII/CSV file, make sure the file meets the following requirements listed in the table below.

Elements in an ASCII/CSV File	Requirement
INTEGER data types	Items of data to be loaded into columns with INTEGER data types cannot contain internal commas unless the item is enclosed in quotes. The default QUOTES character in R:BASE is a single quote (''); if your ASCII file uses double quotes ("), change the QUOTES setting before you load the file. If the file does not have quotes around the integer values containing commas, you must edit the file to remove any commas from the integer values, or enclose each integer value in quotes.
Embedded punctuation	Items of data containing ampersands, commas, embedded blanks, plus signs, equal signs, or semicolons must be enclosed in quotes if they are to be loaded into columns with a TEXT or NOTE data type. The default QUOTES character in R:BASE is a single quote (''); if your file uses double quotes ("), change the QUOTES setting before you load the file.
Embedded quotes	Items of data requiring quotes can also contain embedded quotes. For example, the item <i>'Basic' Keyboard</i> contains both a blank space and embedded quotes. Using single quotes ('), which is the default QUOTES setting, to add enclosing quotes, the item would look like this: <i>'''Basic'' Keyboard'</i>
Currency	R:BASE automatically adds a currency symbol, commas, and zeros for currency units. For example, using the default currency format, R:BASE loads an entry of 1000 as \$1,000.00.
Dates	The SET DATE SEQUENCE command sets the sequence for the date-dates in the file are loaded if the dates match the current date sequence established with the SET DATE command.
Computed columns	If the table being loaded has computed columns and the file contains values for the computed columns, R:BASE tries to load the computed column's value from the file into the column following the computed column. This results in an error because the data type of the next column might not be the correct data type, or the file will have too many values for the table because R:BASE does not load the computed column's value from the file.
Rules processing	Unless you run the SET RULES OFF condition before loading the file, rules processing is in effect. When an incoming data item violates a rule, R:BASE does not load the row. Instead, R:BASE displays the message for the rule that has been violated. To see the data that causes a rule violation, SET ECHO ON when loading a table and use the [Pause] key to stop the screen from scrolling when the rule violation occurs.

Loading a Data Block

The data block shown in the diagram can include lines of data and any of the options available with LOAD-CHECK/NOCHECK, FILL/NOFILL, and NUM/NONUM. You can intersperse the options with data lines,

and you can enter more than one option on a line if you separate the options with semicolons. However, you cannot combine data and options on the same line.

R:BASE displays the dialog prompt to accept data-block entry. LOAD adds data to a table, row by row, without using a data-entry form and without prompting for each data item.

You can enter the options for the LOAD command at the dialog prompt at any time during data loading. Or you can include them on the command line, separated from the command by semicolons, as shown in the example below. (Do not use this format in command or procedure files. All options must follow the LOAD command on separate lines in command or procedure files.)

```
LOAD transdetail ; CHECK ; NUM
```

You can use global or system variables instead of constant values in the data block.

To enter values properly, use the following guidelines.

- Enter column values in the order that columns are defined in the table, and separate the values with a delimiter character. The default delimiter character is the comma.
- You can enter up to 75 characters on a single line. If a row is longer than 75 characters, continue on to the next line by typing past the end of the current line or by entering a plus (+) sign at any point on the current line. The plus sign must be the last entry on the line. The new line will begin with a +> prompt to indicate the continuation of the current line. If you are using this form of the LOAD command in a command file, you must use a + to continue on the next line; the lines will not automatically wrap.
- For other requirements on loading data, see "Loading from an ASCII File" earlier in this entry.

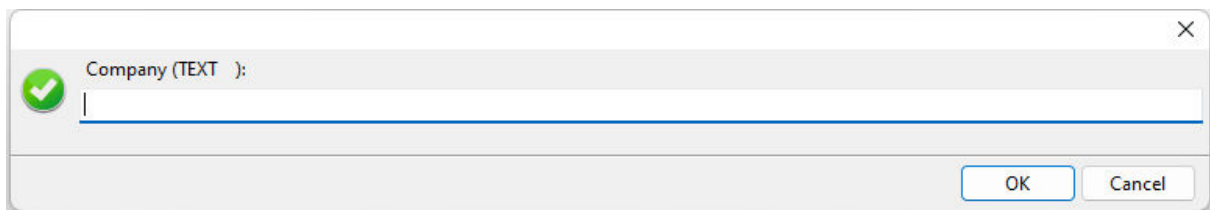
Examples

Example 01:

The following command line allow you to load rows containing new customer information to the *customer* table. R:BASE asks with prompts for each column by column name and data type. Two columns in the table, *custid* and *custphone*, are omitted from the list. R:BASE automatically supplies a number for the *custid* column because it is an autonumbered column. R:BASE leaves the *custphone* column empty (null) when data is loaded, and does not prompt for either column.

```
LOAD customer WITH PROMPTS USING company, +
custaddress, custcity, custstate, custzip
```

After the above command is run, the WITH PROMPTS option displays the message below.



If you press the [Esc] key or the "Cancel" button before you have finished entering data in a row, the row is not added to the table. You will be prompted to add another row.

To continue, press the "Yes" button. To exit, press the "No" button or the [Esc] key.

Example 02:

The following command loads five rows of data into the *customer* table from CUST.DAT, a delimited ASCII file. The data in the ASCII file must be in the same order as the columns in the *customer* table. Only the first five lines from the file will be loaded:

```
LOAD customer FROM cust.dat FOR 5 ROWS
```

Example 03:

In the following example, the command line tells R:BASE to start loading data for the *customer* table. A dialog prompt is displayed for each new row. Each column value would be entered in this one dialog and separated with a comma, or the current delimiter. The length of the text available to fit in the dialog is 4096 characters.

```
LOAD customer
```

Example 04:

After the command line in the following example is run, R:BASE expects the next five lines entered at the dialog prompt to contain data to be loaded into the *customer* table. After the fifth line of data is entered, the loading ends. To end loading before five rows of data are entered, enter END.

```
LOAD customer FOR 5 ROWS
```

Example 05:

The following command lines show you how to load data into the *company*, *custaddress*, *custcity*, *custstate*, and *custzip* columns of the *customer* table. The *custid* and *custphone* columns in the *customer* table will not have data loaded and will be given null values.

```
LOAD customer FROM customer.fix AS FORMATTED +  
USING company 11 50, custaddress 51 80, +  
custcity 81 100, custstate 101 102, custzip 103 112
```

Example 06:

When you use the LOAD command, you must omit values for computed or autonumbered columns. Instead, enter the value for the next column in the data list. In the following example, to add a row to the *transdetail* table, which has a computed column, you would only enter data for the first five columns; the sixth column is a computed column based on the fourth and fifth columns. The columns entered are *transid*, *detailnum*, *model*, *units*, and *price*. The computed column is *extprice* and has the expression $(units * price)$.

```
LOAD transdetail  
6000,1,'CX3000',100,$1900  
END
```

Example 07:

The following loads an image file into a table. The square brackets "[]" tell R:BASE that an image is being loaded, rather than text. The file name within the brackets must be within quotes using the value for the QUOTES setting. The column for the image must be defined as the VARBIT data type.

```
LOAD ImageTable  
1,['image.bmp']  
END
```

1.14 M

1.14.1 MAXIMIZE

Use the MAXIMIZE command to maximize the R> Prompt window. MAXIMIZE was used in earlier Windows versions of R:BASE with MDI. The command will behave as expected, but is no longer needed.

MAXIMIZE

1.14.2 MIGRATE

The MIGRATE command is used to convert prior version databases to R:BASE 11.

MIGRATE dbname

Options

dbname

Specifies the database to be converted.

About the MIGRATE Command

R:BASE 11 requires the conversion of prior version databases for structural changes. Once the database is converted, it CANNOT be accessed by any previous version of R:BASE. Be sure and backup your database before you migrate it.

1.14.3 MINIMIZE

Use the MINIMIZE command to minimize the R> Prompt window. MINIMIZE was used in earlier Windows versions of R:BASE with MDI. The command will behave as expected, but is no longer needed.

MINIMIZE

1.14.4 MKDIR

Use the MKDIR command to create a new directory.

MKDIR [**dirname**]
 [**C:\...\dirname**]

Options

pathname

Specifies one or more directory names that lead from the root directory to a specified directory.

About the MKDIR Command

The R:BASE MKDIR command is similar to the operating system MKDIR command. For more information about the MKDIR command, refer to your operating system manual.

Examples

The following command creates the OLDDATA directory as a sub-directory in the current directory.

```
MKDIR olddata
```


1.15 N

1.15.1 NEWDB

The NEWDB command is used to convert database files to the current version format. NEWDB is specifically able to convert legacy database formats including those with .RBF and .RB1-.RB3 file extensions. NEWDB will convert the database to the .RX1-.RX4 format.

```
NEWDB dbname [ IDENTIFIED BY [ OWNER ] ]
```

Options

dbname

Specifies the database to be converted.

IDENTIFIED BY

Specifies the user identifier. If left blank, R:BASE prompts you for the user identifier. R:BASE does not display it as you enter the text.

OWNER

Optional; specifies the database owner name. If omitted and an OWNER name exists, you will be prompted.

About the NEWDB Command

R:BASE requires the conversion of existing legacy database in order to connect to the files. Once the database is converted, it CANNOT be accessed by any previous version of R:BASE. Be sure and backup your database before you convert it.

If there is illegal use of table and column names (containing special characters) a DBName.ERR file will be created with the list of items to address.

After the database conversion, users must still convert forms and reports using the Database Explorer.

1.15.2 NEWPAGE

Use the NEWPAGE command to send a form feed to the current output device: printer, screen, or file.

```
NEWPAGE
```

About the NEWPAGE Command

When the current output device is the computer screen, entering the NEWPAGE command clears the window and moves the R> Prompt and cursor to the top of a new window. When the current output device is a printer, entering the NEWPAGE command sends a form feed that advances the paper to the top of the next page. When the current output device is a file, entering the NEWPAGE command sends a form feed character to the file.

1.15.3 NEWROW

Use the NEWROW command in an entry/exit procedure (EEP) to add a row.

```
NEWROW
```

About the NEWROW Command

The NEWROW command displays a blank data entry form for the user to enter data.

To see the effect of the NEWROW command, you must include the [SCREEN RESTORE OFF](#) command in the EEP.

1.15.4 NEXTROW

Use the NEXTROW command in an entry/exit procedure (EEP) to move to the next row.

NEXTROW

1.15.5 NEXTTAB

Use the NEXTTAB command in an entry/exit procedure (EEP) to move to the next table.

NEXTTAB

1.15.6 NORMALIZE

Use the NORMALIZE command to size the R> Prompt window to the default size. NORMALIZE was used in earlier Windows versions of R:BASE with MDI. The command will behave as expected, but is no longer needed.

NORMALIZE

1.15.7 NULLSET

Use the NULLSET command to set variable values to NULL for individual variables, or a list variables.

NULLSET { **VARIABLES varlist**
ALL VARIABLES [**EXCEPT varlist**] }

Options

ALL VARIABLES

Sets all variables to NULL.

EXCEPT varlist

Specifies variable values that the NULLSET command will not set to NULL. You can use wildcards in variable names.

VARIABLES varlist

Sets one or more variable values to NULL. Use this option at the end of a complete set of procedures to set variable values to NULL. You can use wildcards in variable names.

About the NULLSET Command

NULLSET provides the ability to NULL variable values for a list variables. Only system variables are not effected by the NULLSET command.

Examples

Example 01:

-- Sets the *vCost* and *vReportName* variable values to NULL

```
NULLSET VAR vCost, vReportName
```

Example 02:

-- Sets the all variable values that begin with *vcust* to NULL

```
NULLSET VAR vCust%
```

Example 03:

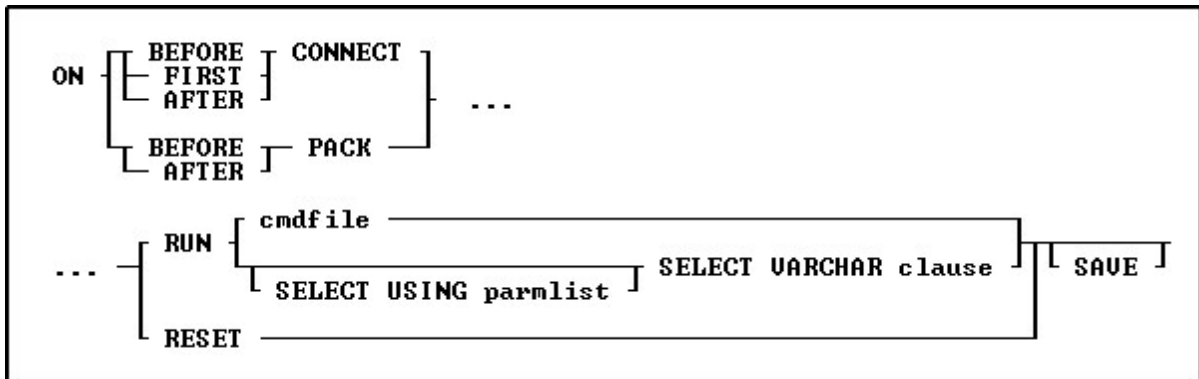
-- Sets all variable values to NULL, with the except of system variables

```
NULLSET ALL VAR EXCEPT vOutput
```

1.16 O

1.16.1 ON CONNECT/PACK

Use the ON CONNECT/PACK command to run a command file before or after connecting to or packing a database.



Options

AFTER

Indicates the command file should be run after the CONNECT/PACK.

BEFORE

Indicates the command file should be run before the CONNECT/PACK.

CONNECT

Indicates the command file should be run when connecting a database.

FIRST

Indicates the command file should be run on the first connect only.

PACK

Indicates the command file should be run when packing a database.

RESET

Clears out the ON CONNECT/PACK command recorded in the database.

RUN

Runs a command file or procedure stored within a table.

cmdfile

Specifies the name of the command file to execute.

SAVE

Specifies that the parameters in the **cmdfile** or from the **RESET** will be saved to the database. Any options without the SAVE parameters will only be active for a session.

SELECT VARCHAR clause

Specifies a column defined with the VARCHAR data type from a table, from which you can run the contents. The [SELECT](#) clause must limit the data to only one row; otherwise, an error is returned.

USING parmlist

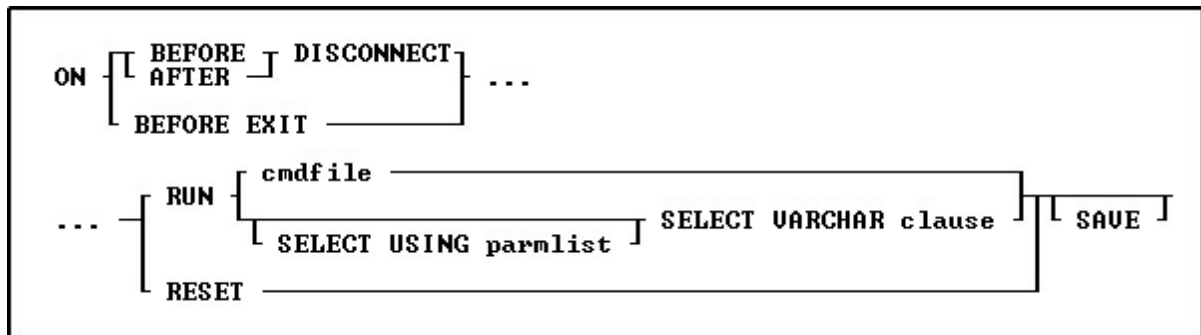
Lists the values the command file uses when it runs. The parameter list can contain up to 18 values. The first value in the list is referenced in the executed file as %1, the second as %2, and so on through %9. They are treated just like other variables. To reference the contents of these variables, preface the variable name with a dot (.); for example, set v1 =.%1.

About the ON CONNECT/PACK command

The ON CONNECT/PACK command is useful for running a specific command file whenever a database is connected or packed. A common use would be to run a command file that does an [AUTOCHK](#) command before connecting. The [ABORT ON](#) command could then be used to cause the CONNECT/PACK to be aborted after the command file finishes.

1.16.2 ON DISCONNECT/EXIT

Use the ON DISCONNECT/EXIT command to run a command file before or after disconnecting from a database or closing the R:BASE session.

**Options****AFTER**

Indicates the command file should be run after the DISCONNECT.

BEFORE

Indicates the command file should be run before the DISCONNECT.

BEFORE [EXIT](#)

Indicates the command file should be run before exiting R:BASE.

[DISCONNECT](#)

Indicates the command file should be run when disconnecting from database.

RESET

Clears out the ON DISCONNECT/EXIT command recorded in the database.

RUN

Runs a command file or procedure stored within a table.

cmdfile

Specifies the name of the command file to execute.

SAVE

Specifies that the parameters in the **cmdfile** or from the **RESET** will be saved to the database. Any options without the SAVE parameters will only be active for a session.

SELECT VARCHAR clause

Specifies a column defined with the VARCHAR data type from a table, from which you can run the contents. The [SELECT](#) clause must limit the data to only one row; otherwise, an error is returned.

USING parmlist

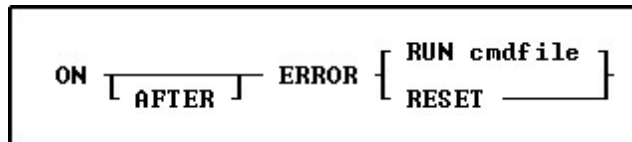
Lists the values the command file uses when it runs. The parameter list can contain up to 18 values. The first value in the list is referenced in the executed file as %1, the second as %2, and so on through %9. They are treated just like other variables. To reference the contents of these variables, preface the variable name with a dot (.); for example, set v1 =.%1.

About the ON DISCONNECT/EXIT command

The ON DISCONNECT/EXIT command is useful for running a specific command file whenever a database is disconnected or the R:BASE session is closed. An example would be to run a command file after disconnecting to reinforce desired settings. The ON EXIT command can be used for logging events within custom transaction tables.

1.16.3 ON ERROR

Use the ON ERROR command to execute a command after an error occurs.

**Options****AFTER**

Indicates the command file should be run after the ERROR. The use of the AFTER keyword can be omitted for backwards compatibility, but should be used in current releases.

RUN cmdfile

Specifies the name of the command file to execute.

RESET

Turns ON AFTER ERROR processing off.

About the ON ERROR Command

The default is ON AFTER ERROR RESET.

Any command that can be used in an entry/exit procedure, or EEP, can be used with ON AFTER ERROR except: form specific commands such as [NEXTROW](#), [DUPROW](#), [SKIP](#), etc. If a command cannot be used in an EEP, such as [RBDEFINE](#), [RESTORE](#), [RBAPP](#), and [RBSYNC](#), then it cannot be used with ON AFTER ERROR.

The command file content must be limited to 2000 characters. To review the character count for a file within R:BASE Editor, choose "File" > "Information" from the main menu bar.

Since this command remains in effect until it is re-issued, you should place `ON AFTER ERROR RESET` at the end of command files that use it. If `ON AFTER ERROR` runs a command file, the file must end with a [RETURN](#) command.

Example:

```
ON AFTER ERROR RUN ErrorLog.rmd
```

1.16.4 OPEN

Use the `OPEN` command before using the cursor designated by the [DECLARE CURSOR](#) command.

OPEN <i>cursorname</i> [RESET]
--

Options

cursorname

Specifies a 1 to 18 character cursor name that has been previously specified by the `DECLARE CURSOR` command.

RESET

Reopens a cursor with the current values of any variables referenced in the `DECLARE CURSOR` statement. This improves performance by eliminating the need to re-optimize the query.

About the `OPEN` Command

`OPEN` evaluates the [SELECT](#) clause of the [DECLARE CURSOR](#) command using the current values of any variables that it contains. Then `OPEN` stores that copy of the cursor definition and places the cursor before the first row.

After you close a cursor with the [CLOSE](#) command, you can reopen it by repeating the `OPEN` command. Every time you open a cursor, R:BASE reads the rows again, so that any changes you previously made through the cursor are visible when you look at the rows.

You can display all currently defined cursors with the [LIST CURSOR](#) command.

When using the `RESET` option, the `WHERE` clause is evaluated with the current values of any referenced variables, and the cursor is reopened without requiring a `CLOSE` command. The cursor is positioned at the beginning of the result set when the [FETCH](#) command is run.

Example

The following command lines show the `OPEN` command with the `RESET` option.

```
DROP CURSOR c1
DROP CURSOR c2
SET VAR vclcustid INTEGER
DECLARE c1 CURSOR FOR SELECT custid FROM customer
-- Selects the transaction rows for the customer
DECLARE c2 CURSOR FOR SELECT transid, invoicetotal +
    FROM transmaster WHERE custid = .vclcustid
-- Process the query in cursor c1 and get the first custid
OPEN c1
FETCH c1 INTO vclcustid IND clind1
WHILE sqlcode <> 100 THEN
    -- Process the query in c2. As each row is fetched in the
    -- customer table, the custid changes; each time
    -- the "OPEN c2 RESET command" processes the c2's query it
```

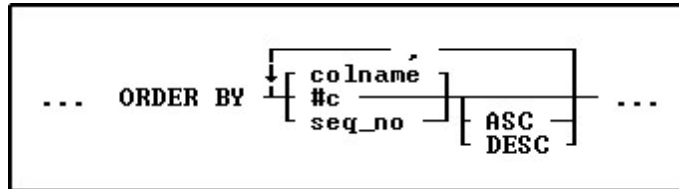
```

-- retrieves different rows
OPEN c2 RESET
-- Fetch the transid (invoicenumber) and invoice total amount
FETCH c2 into vtransid, vamt
WHILE sqlcode <> 100 THEN
    WRITE .vtransid, .vamt
    FETCH c2 into vtransid, vamt
ENDWHILE
FETCH c1 INTO vclcustid
ENDWHILE
CLOSE c1
CLOSE c2
DROP CURSOR c1
DROP CURSOR c2

```

1.16.5 ORDER BY

Use the ORDER BY clause with an R:BASE command to specify the order in which rows of data from a table are displayed.



Options

,

Indicates that this part of the command is repeatable.

ASC DESC

Specifies whether to sort a column in ascending or descending order.

#c

Takes the place of a column name and refers to the column numbers displayed with the [LIST TABLE](#) command.

colname

Sorts by any column name or combination of column names.

seq_no

Refers to the items listed in the [SELECT](#) command that is using the ORDER BY command, ordered from left to right. An item can be a column name, expression, or SELECT function.

About the ORDER BY Command

The syntax for the ORDER BY clause is the same for all commands. ORDER BY must refer to only one table or view.

You can significantly reduce the time R:BASE takes to process an ORDER BY clause when the column or columns listed in the ORDER BY clause are included in an index with the same column sort order as that specified in the ORDER BY clause.

Using the SET SORT Command

The ORDER BY command uses the R:BASE automatic sort optimizer. If you are sorting extremely large tables, and if your disk space is limited, the automatic sort optimizer might be unable to sort the data. Instead, use the [SET SORT ON](#) command because it uses the least disk space necessary to sort data; however, the SET SORT ON command is slower than the automatic sort.

Examples

The following command displays data from the *custid*, *company*, and *custcity* columns from the *customertable*.

```
SELECT custid, company, custcity FROM customer
```

The ORDER BY clause in the command below arranges the *custid* values in descending order.

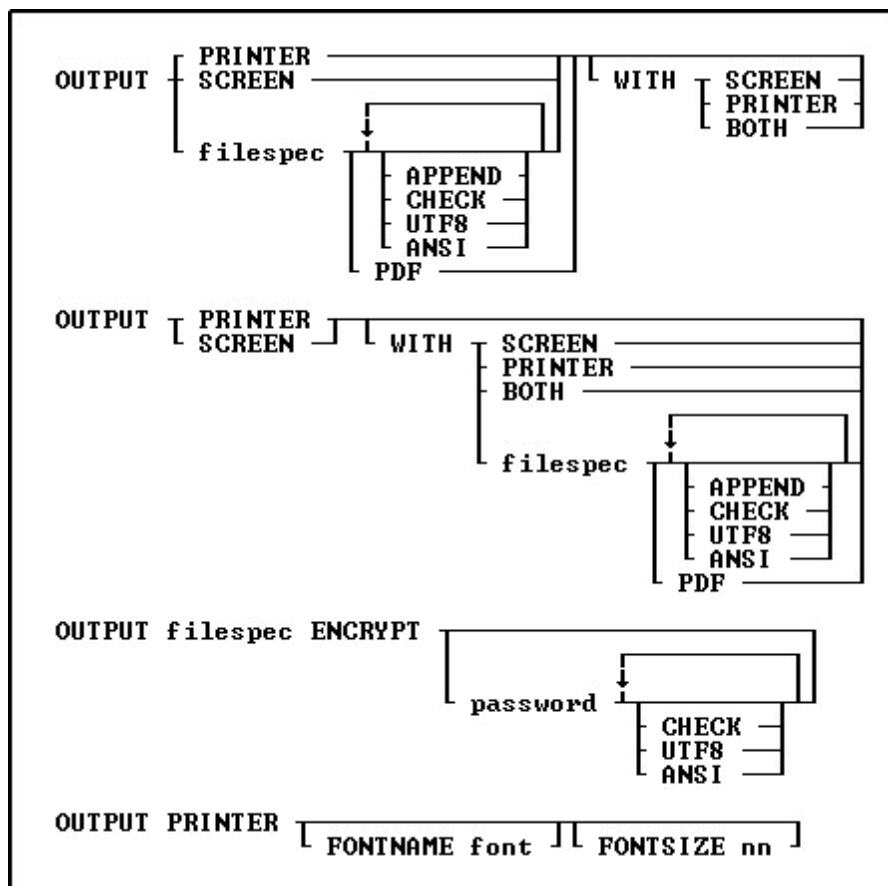
```
SELECT custid, company, custcity FROM customer +  
ORDER BY custid DESC
```

You can substitute a column's sequence number for a column named in the ORDER BY clause. You must use a sequence number when referring to an expression, function, constant, or when a UNION operator is used. The following command is equivalent to the command example above.

```
SELECT custid, company, custcity FROM customer +  
ORDER BY 1 DESC
```

1.16.6 OUTPUT

Use the OUTPUT command to direct messages and results of commands to a file, printer, screen, or a combination of destinations.



Options

ANSI

Converts all UTF8 characters to ANSI, to use the output in programs that do not understand UTF8 characters, but does handle the ANSI characters with code values above 127. ANSI and UTF8 would not be used simultaneously.

APPEND

Appends data to the end of an existing file without overwriting the file. If you specify APPEND when the specified file does not exist, R:BASE creates the file with that name. APPEND and CHECK would not be used simultaneously.

BOTH

Refers to PRINTER and SCREEN.

CHECK

Checks for file existence and prompts the user for confirmation before writing to it. APPEND and CHECK would not be used simultaneously.

ENCRYPT

A 512-bit encryption method is used to obscure any output information, making it unreadable without R:BASE and your decryption password. Immediately after using the ENCRYPT parameter in your OUTPUT command, you will be prompted for a password. The password is limited to 32 characters. When running encrypted files with R:BASE, you would [RUN](#) the file followed by the password. When opening an encrypted file, you will be prompted for the password.

filespec

Indicates the output device. Specify a file name, with or without an extension. You can also specify a drive and/or path.

FONTNAME font

Specifies a custom font for the printer output. The default font is Courier New.

FONTSIZE nn

Specifies a custom font size for the printer output. The default font size 10.

password

Specifies the password for the encrypted file. If the password is not specified, a prompt to enter the password will be displayed.

PDF

Creates the output as a PDF file. The filespec parameter should use the PDF file extension.

The PDF output uses Courier as the default font. The PDF output will use the "Default Printer Font" settings with the following PROPERTY command:

```
PROPERTY APPLICATION OUTPUT_PDF_USE_PRINTER_FONT ON
```

The PDF output can also be configured to use default printer's paper size with the following PROPERTY command:

```
PROPERTY APPLICATION OUTPUT_PDF_USE_PRINTER_DEF_PAPER ON
```

PRINTER/Printer

Indicates the output device. *Printer* is either the word PRINTER or a valid output port, such as LPT1 or COM1. An output port is treated as a file that is automatically copied to the printer device when an OUTPUT SCREEN command is issued.

SCREEN

Indicates the output device is the screen.

UTF8

Adds a UTF8 BOM to the front of the output file. ANSI and UTF8 would not be used simultaneously.

About the OUTPUT Command

The default output device is the screen. After sending data to a printer or file, change the output device to the screen. When using the `OUTPUT filespec` command, you might lose the data in the file if you exit from R:BASE without sending an `OUTPUT SCREEN` command to close the file.

You cannot combine two *filespec* designations such as `OUTPUT filespec WITH filespec`.

R:BASE cannot tell that the argument LPT1 or COM1 is not the name of a file. For example, if you use the command `OUTPUT LPT1`, R:BASE creates a file named LPT1. Because R:BASE thinks LPT1 is a file, R:BASE does not print the last buffer of output until you redirect output to the screen.

Sending Data to a Printer

Printer refers to the system default printer. In Windows, to change a printer, use the [PRNSETUP](#) command or the Windows Control Panel. In DOS, *Printer* refers to the printer that is attached to LPT1.

If you are using R:BASE on a local area network, you can direct output to any printer to which you have network access. A common method is to enter the `OUTPUT` command followed by either the network name of the printer or the printer port on the server.

Sending Data to the Screen

The [SET LINES](#) command determines how many lines display on a screen. If a file contains more lines than are designated to display on a screen, R:BASE pauses after each screen display. When displaying a report on a screen, the number of lines specified in the report are displayed.

If the primary output device is a file or a printer and the secondary output device is the screen, R:BASE pauses after each screen display.

Backing Up with OUTPUT

The `OUTPUT` command can be used with the [BACKUP](#) or [UNLOAD](#) commands to backup your database to a file. If some part of the command is incorrect—for example, if you fail to specify the correct user identifier for the database—the resulting file contains only the error message produced by the incorrect entry. Use the [TYPE](#) or [DISPLAY](#) command to display the contents of the file and view the results.

Using SET ECHO ON

To send both your commands and the results of the commands to an output device, enter the command [SET ECHO ON](#) before beginning the `OUTPUT` session. The `SET ECHO OFF` command only sends command results to an output device.

Examples

Example 01:

The `OUTPUT` command sends data and messages to the printer and simultaneously displays the data and messages on screen with pauses between screens, if needed.

```
OUTPUT PRINTER WITH SCREEN
```

Example 02:

The `OUTPUT` command directs output to the `BACKUP.DAT` file on drive A:\. The `UNLOAD` command sends the data stored in the *TransMaster* table to the file. The second `OUTPUT` command closes the file and redirects output to the screen.

```
OUTPUT a:\backup.dat
UNLOAD DATA FOR TransMaster
OUTPUT SCREEN
```

Example 03:

The command sends data and messages to the printer, designated as COM1. COM1 is treated as a file that is automatically copied to a printer device when the `OUTPUT SCREEN` command is issued.

```
OUTPUT COM1
```

Example 04:

The commands send data from the *Customer* table to the printer using a larger font size.

```
OUTPUT PRINTER FONTSIZE 12
```

```
SELECT Company, CustCity, CustState FROM Customer
OUTPUT SCREEN
```

Example 05:
The commands send data from the Customer table to the printer using the Arial 7 font.

```
OUTPUT PRINTER FONTNAME Arial FONTSIZE 7
SELECT Company, CustCity, CustState FROM Customer
OUTPUT SCREEN
```

Example 06:
The commands send data from the Customer table to a PDF file, and open the file with the [LAUNCH](#) command.

```
OUTPUT CustomerList.PDF PDF
SELECT Company FROM Customer ORDER BY Company
OUTPUT SCREEN
LAUNCH CustomerList.PDF
```

Example 07:
The commands unload the structure and data from the database into an encrypted file, and checks that the file does not already exist

```
OUTPUT DB_ALL_BACKUP.ALL ENCRYPT pw1234 CHECK
UNLOAD ALL
OUTPUT SCREEN
```

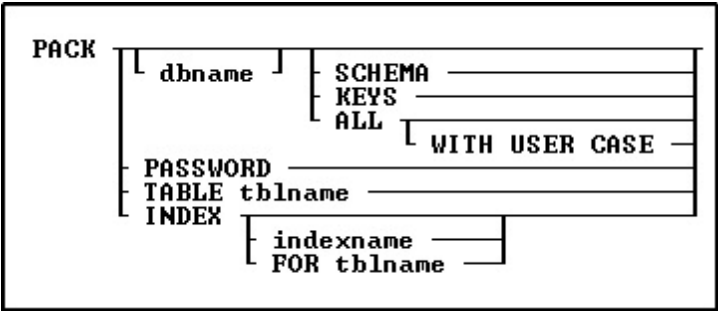
Example 08:
The commands unload the data from the Customer table with ANSI file encoding

```
OUTPUT Customer.dat ANSI
UNLOAD DATA FOR Customer
OUTPUT SCREEN
```

1.17 P

1.17.1 PACK

Use the PACK command on an open database to recover unusable disk space.



Options

ALL
Packs File 1 (schema information), File 2 (data), File 3 (indexes), and File 4 (large object data).

dbname
Specifies the name of the database to pack.

FOR tblname

Specifies a particular table whose indexes you want to PACK.

INDEX

Use this option to PACK all indexes for the currently connected database. PACK INDEX is supported in multi-user environments.

indexname

Packs a specified index from File 3 (indexes); this option will work when [STATICDB](#) is set on. PACK INDEX indexname is now supported in multi-user environments. This command will execute on the currently connected database.

KEYS

Packs only File 3 (indexes).

PASSWORD

Use this option to remove deleted rows from the SYS_PASSWORDS table. A "PACK TABLE SYS_PASSWORDS" is performed upon the system table. The completion of the PACK PASSWORD command will also assign the correct row count to the SYS_PASSWORDS table. The PACK PASSWORD command is supported in a multi-user session. The database must be connected in order to use this command.

SCHEMA

Packs only File 1 (schema information). The database schema stores the internal order in which tables and views were defined. When the database structure is [unloaded](#), the order is used for primary key tables to be created before foreign key tables. Views are also stored in the order of creation, where views based on other views are unloaded after all the other views it is dependent upon.

If any view that another view it is dependent upon is deleted and redefined, a PACK SCHEMA should be performed. Then, the view with the dependencies should be deleted and recreated. This would place the view with the dependencies after all the other views it is dependent upon within the internal order. The steps would ensure a database UNLOAD and rebuild is successful where views with the dependencies are created last.

TABLE tblname

Packing a single table when [MULTI](#) is set ON. This parameter is very beneficial with databases that are always is use.

WITH USER CASE

Replaces the case folding/collating tables in the database with those defined in the user configuration file.

About the PACK Command

Disk space becomes unusable when you delete rows or indexes, remove columns or tables, or add or modify columns with the [ALTER TABLE](#) command. To use the PACK command, a database must be open. If the database you want to pack is not in the current directory, include the drive, path, and database name. When you pack a database that is in a different directory, R:BASE closes any open database, then opens the database you want to pack.

[PACK](#) requires the database owner's user identifier if the database is protected by the owner's user identifier.

You cannot use PACK when a database is stored on a network drive and [MULTI](#) has been set on, unless you are ONLY packing one table. Set MULTI off before packing the database.

PACK is unavailable when transaction processing is on.

Because you pack an open database, backup your database before you pack it. An interruption to a pack could cause damage to your database.

PACK Versus RELOAD

Both the [PACK](#) and [RELOAD](#) commands recover unusable disk space; however, RELOAD requires more disk space than PACK because RELOAD copies a database table by table, collects the rows of each table, then reorganizes the rows on the disk.

PACK KEYS versus PACK INDEX

PACK KEYS is to be used with [MULTI](#) set OFF as it recreates a new index file with clean indexes. PACK INDEX can be used while users are connected to the database with MULTI set ON, only adding to the current index file. After using PACK KEYS, you should see a decrease in the index file size.

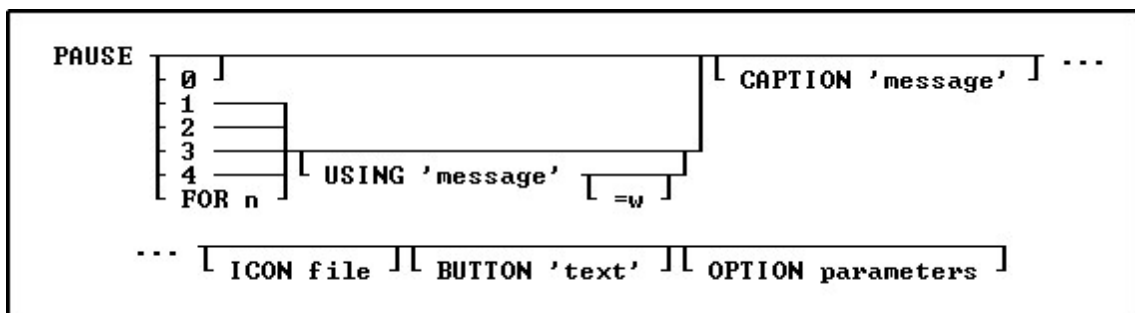
Example

The following command packs the *ConComp* database in the RBTI directory on drive C:

```
PACK C:\RBTI\ConComp
```

1.17.2 PAUSE

Use the PAUSE command to display a message and can also suspend the running of the command file. For assistance with building your PAUSE commands, refer to the [R:Pause Builder Plugin](#).

**Options****0**

Pauses without a message.

1

Displays "Press OK to continue" or any message using the USING option in a dialog box. The R> Prompt Input Console and Output Console are then cleared with the next keystroke.

2

Same as the 1 option, except the R> Prompt screen does not clear with the next keystroke.

3

Does not pause the running of the command file and therefore does not wait for the next keystroke.

4

If a PAUSE 3 dialog already exists only the message will be repainted to avoid flickering, when not using additional OPTIONS parameters. Otherwise, this is exactly like the PAUSE 3 option. When using OPTIONS parameters in a PAUSE 3, and then following the code with a PAUSE 4, the OPTIONS parameters must be included in the PAUSE 4 command syntax as well.

BUTTON 'text'

Specifies the text of the button. The value can also be passed as a variable.

FOR n

Sets the pause duration in seconds; **n** must be a positive integer. Any keystroke interrupts the pause, regardless of duration. In the absence of a USING clause, no message is displayed. With a USING clause, the message is displayed in a dialog box. When using subsequent PAUSE FOR n USING commands, the [CLS](#) command is needed between the PAUSE commands to ensure optional parameters do not carry over from one dialog to the next.

USING 'message'

Displays the specified message in a dialog box. This value can also be passed as a variable.

Using this default parameter the message text is limited to one line. However, if you need to display a multi-line PAUSE window, you can create separate lines with the ASCII characters for a carriage return and indent ([Tab] key). A sample is provided below.

=w

Specifies the wrap width for the message.

CAPTION 'message'

Specifies the text of the message to display in the window caption. The value can also be passed as a variable.

ICON *value*

Icon " <i>value</i> " Parameter	Icon
APPS	
ATTENTION	
CONFIRM	
ERROR	
HELP	
INFO	
QUESTION	
SERIOUS	
STOP	
WARNING	
WINDOWS	

* will vary based upon Windows operating system

Additional [OPTION parameters](#)

Additional parameters are available to increase the visual display of the PAUSE window. To use the graphic PAUSE Builder, choose "Utilities" > "Plugins" > "PAUSE Builder" from the main Menu Bar. All OPTION parameters and values must be separated by the "|" (pipe) character.

OPTION Parameters	Description
Message	Changes how the PAUSE "message" is displayed
Button	Changes how the PAUSE "button" are displayed
Meter	Displays a "meter" progress bar in the PAUSE
Gauge Meter	Displays a "gauge meter" in the PAUSE

CMPAUSE Setting

In instances where several PAUSE dialogs will appear, perhaps within a loop, the [CMPAUSE](#) setting is available to display the messages in a cascade modal mode.

Examples:

Example 01: (PAUSE with an ICON)

```

PAUSE 2 USING 'Message Text' +
CAPTION 'Caption Text' +
ICON HELP OPTION +
BUTTON 'Button Text' +
|BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_COLOR GREEN +
|BUTTON_COLOR WHITE +
|BUTTON_FONT_COLOR GREEN +
|TRANSPARENCY 255

```

Example 02: (PAUSE with ICON and custom [message](#) format)

```

PAUSE 2 USING 'Message Text' +
CAPTION 'Caption Text' +
BUTTON 'Button Text' +
OPTION ICON_FILE path\directory\filename.bmp +
|BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_COLOR GREEN +
|BUTTON_COLOR WHITE +
|BUTTON_FONT_COLOR GREEN +
|TRANSPARENCY 255

```

Example 03: (Multi-Line PAUSE)

```

-- (CHAR(009)) = Tab Key (Indent)
-- (CHAR(013)) = Carriage Return

SET VAR vMsg = +
('Line 1:'+(CHAR(009))+(CHAR(009))&'Contents of Line 1'+(CHAR(009))+(CHAR(013))+ +
'Line 2:'+(CHAR(009))+(CHAR(009))&'Contents of Line 2'+(CHAR(009))+(CHAR(013))+ +
'Line 3:'+(CHAR(009))+(CHAR(009))&'Contents of Line 3'+(CHAR(009))+(CHAR(013))+ +
'Line 4:'+(CHAR(009))+(CHAR(009))&'Contents of Line 4'+(CHAR(009))+(CHAR(013))+ +
'Line 5:'+(CHAR(009))+(CHAR(009))&'Contents of Line 5'+(CHAR(009))+(CHAR(013))+ +
'Line 6:'+(CHAR(009))+(CHAR(009))&'Contents of Line 6'+(CHAR(009))+(CHAR(013))+ +
'Line 7:'+(CHAR(009))+(CHAR(009))&'Contents of Line 7'+(CHAR(009))+(CHAR(013))+ +
'Line 8:'+(CHAR(009))+(CHAR(009))&'Contents of Line 8'+(CHAR(009))+(CHAR(013))+ +
'Line 9:'+(CHAR(009))+(CHAR(009))&'Contents of Line 9'+(CHAR(009))+(CHAR(013)))

PAUSE 2 USING .vMsg +
CAPTION 'Caption Text' +
ICON APP +
BUTTON 'Button Text' +
OPTION BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_COLOR GREEN +
|BUTTON_COLOR WHITE +
|BUTTON_FONT_COLOR GREEN +
|TRANSPARENCY 255

```

Example 03: (PAUSE with Themes)

```

PAUSE 2 USING +
'Now you can add themes to PAUSE windows!' +
CAPTION 'New PAUSE Command' ICON INFO +
OPTION THEMENAME Longhorn

```

Example 04: (PAUSE with [Meter](#) Progress Bar)

```

PAUSE 3 USING +
'PAUSE Command Text with More OPTIONS - Imagine the Possibilities!' +
CAPTION ' ' +
OPTION METER_VISIBLE ON +
|METER_VALUE .vProgress +
|METER_TYPE BAR3D +
|METER_BACK_COLOR WHITE +
|METER_BAR_COLOR NAVY +
|METER_FONT_COLOR RED +
|BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_COLOR GREEN

```

Example 05: (PAUSE with [GAUGE](#) Progress Bar)

```

PAUSE 3 USING 'Calculating ... Please Stand By ...' +
CAPTION '          Pause 3 with Gauge ' +
ICON WINDOWS  OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR RED +
|GAUGE_INTERVAL 10 +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR WHITE +
|THEMENAME Steel Blue

```

Example 06: (PAUSE with Custom [Button](#))

```

PAUSE 2 USING 'You MUST Enter Last Name!' +
CAPTION 'Your Caption Text Here ...' +
ICON INFO +
BUTTON 'Click here to continue ...' +
OPTION BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_COLOR -16777216 +
|BUTTON_FONT_NAME Tahoma +
|BUTTON_FONT_COLOR -16777208 +
|BUTTON_FONT_SIZE 10 +
|BUTTON_FONT_BOLD ON +
|BUTTON_WIDTH 220 +
|BUTTON_HEIGHT 35

```

1.17.2.1 OPTION parameters

These parameters can be used to change the "look and feel" of a PAUSE window. In addition to the parameters below, you can also alter the PAUSE [Message](#) and [Button](#), as well as add a [Meter](#) or [Gauge](#) [Meter](#).

Parameter	Value	Description
ICON_FILE	value	Specifies the external path and name for an image file. NOTE: Any of the below image file formats can be used. The recommended size is 32x32, otherwise the image will be automatically resized to 32x32. To display custom BMP images with transparency, use the first pixel, which

		<p>is on the bottom left corner, as the transparent color. Any pixel with that color becomes transparent.</p> <table><tr><td>wmf</td><td>Windows Metafiles</td></tr><tr><td>emf</td><td>Enhanced Metafiles</td></tr><tr><td>ico</td><td>Icon Files</td></tr><tr><td>bmp</td><td>Windows Bitmap Files</td></tr><tr><td>png</td><td>Portable Network Graphics</td></tr><tr><td>jpeg, jpg</td><td>JPEG Image File</td></tr><tr><td>tif, tiff</td><td>TIFF Image File</td></tr><tr><td>gif</td><td>Compuserve GIF Image</td></tr><tr><td>jp2, j2k, jpc</td><td>JPEG 2000 Image</td></tr><tr><td>tga, afi</td><td>Truevision Targa File</td></tr><tr><td>vst</td><td>Truevision Vista File</td></tr><tr><td>pcx, pcc</td><td>ZSoft PC Paintbrush</td></tr><tr><td>dcx</td><td>DCX Image File</td></tr></table>	wmf	Windows Metafiles	emf	Enhanced Metafiles	ico	Icon Files	bmp	Windows Bitmap Files	png	Portable Network Graphics	jpeg, jpg	JPEG Image File	tif, tiff	TIFF Image File	gif	Compuserve GIF Image	jp2, j2k, jpc	JPEG 2000 Image	tga, afi	Truevision Targa File	vst	Truevision Vista File	pcx, pcc	ZSoft PC Paintbrush	dcx	DCX Image File
wmf	Windows Metafiles																											
emf	Enhanced Metafiles																											
ico	Icon Files																											
bmp	Windows Bitmap Files																											
png	Portable Network Graphics																											
jpeg, jpg	JPEG Image File																											
tif, tiff	TIFF Image File																											
gif	Compuserve GIF Image																											
jp2, j2k, jpc	JPEG 2000 Image																											
tga, afi	Truevision Targa File																											
vst	Truevision Vista File																											
pcx, pcc	ZSoft PC Paintbrush																											
dcx	DCX Image File																											
BACK_COLOR	value	Changes the color of the WINDOW BACKGROUND area. User can specify the integer value or type one of the predefined values.																										
TRANSPARENCY	0 - 255	Set the transparency to specify a layered window with a translucent color. The transparency value specifies the degree of translucency.																										
TOP	value	Moves the pause box location, in pixels, from the top of the screen down.																										
LEFT	value	Moves the pause box location, in pixels, from the left of the screen over.																										
THEMENAME	value	<p>Specifies one of 86 pre-defined Themes, or a custom Theme loaded into R:BASE. See Themes.</p> <p>NOTE: All previously defined PAUSE commands which are redefined to use Themes should be thoroughly checked prior to putting into production as objects and text may look substantially different.</p>																										
NO_FOCUS		Specifies the pause will not be focused when displayed, which can be used to possibly prevent an interruption in the focus transition in an active form.																										

1.17.2.1.1 Message

These parameters alter the text message within the PAUSE window.

Parameter	Value	Description
MESSAGE_COLOR	value	Changes the color of the message text.
MESSAGE_FONT_COLOR	value	Changes the font color of message text.
MESSAGE_FONT_NAME	value	Specifies a font name for the message text. Values would consist of the fonts available on the computer.
MESSAGE_FONT_SIZE	value	Specifies the font size (integer value) of the message text.
MESSAGE_FONT_BOLD	ON OFF	Specifies if the message font is bold type.
MESSAGE_FONT_ITALIC	ON OFF	Specifies if the message font is italic type.
MESSAGE_FONT_UNDERLINE	ON OFF	Specifies if the message font is underline type.
MESSAGE_FONT_STRIKEOUT	ON OFF	Specifies if the message font is strikeout type.
MESSAGE_SHOW_ACCELERATOR_CHAR	ON OFF	Specifies if the message displays the accelerator character. The default value is ON.

```
-- Example (PAUSE with accelerator character OFF)
PAUSE 2 USING +
'The order is complete Fenwick & Sons Glass.' +
CAPTION 'New Sales Order' ICON INFO +
OPTION BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 11 +
|BUTTON_COLOR WHITE +
|BUTTON_FONT_COLOR GREEN +
|MESSAGE_SHOW_ACCELERATOR_CHAR OFF
```

1.17.2.1.2 Button

These parameters alter the button within the PAUSE window.

Parameter	Value	Description
BUTTON_COLOR	value	Changes the background color of the button.
BUTTON_FONT_COLOR	value	Changes the font color of the button.
BUTTON_FONT_BOLD	ON OFF	Specifies if the button font is bold type.
BUTTON_FONT_ITALIC	ON OFF	Specifies if the button font is italic type.
BUTTON_FONT_NAME	value	Specifies the font name for button text.
BUTTON_FONT_SIZE	value	Specifies the font size (integer value) for button text.
BUTTON_FONT_STRIKEOUT	ON OFF	Specifies if the button font is strike out type.
BUTTON_FONT_UNDERLINE	ON OFF	Specifies if the button font is underline type.
BUTTON_WIDTH	value	Specifies the button width in pixels.
BUTTON_HEIGHT	value	Specifies the button height in pixels.

```
-- Example (PAUSE with Custom Button)
PAUSE 2 USING 'You MUST Enter Last Name!' +
CAPTION 'Your Caption Text Here ...' +
ICON INFO +
BUTTON 'Click here to continue ...' +
OPTION BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 12 +
|BUTTON_COLOR -16777216 +
|BUTTON_FONT_NAME Tahoma +
|BUTTON_FONT_COLOR -16777208 +
|BUTTON_FONT_SIZE 10 +
|BUTTON_FONT_BOLD ON +
|BUTTON_WIDTH 220 +
|BUTTON_HEIGHT 35
```

1.17.2.1.3 Meter

These parameters will display a meter within the PAUSE window.

Parameter	Value	Description
METER_VISIBLE	ON OFF	Displays a meter to show a progress in percentage values.
METER_TYPE	NORMAL BAR3D	Three meter options are available.

	LED	
METER_VALUE	0 to 100	Specifies the percentage to display the meter progress.
METER_BAR_COLOR	value	Specifies the color of the meter bar.
METER_BACK_COLOR	value	Specifies the meter background color.
METER_FONT_COLOR	value	Specifies the color of the meter font.
METER_MAX_VALUE	0 to 100	Specifies the maximum percentage to display the meter progress.
METER_COLOR1	value	Specifies the color of the meter when using three color segments (green, yellow, red)
METER_COLOR2	value	Specifies the color of the meter when using three color segments (green, yellow, red)
METER_COLOR3	value	Specifies the color of the meter when using three color segments (green, yellow, red)
METER_COLOR_OFF		Sets the color off when using three color segments.

Example:

```
PAUSE 3 USING +
'PAUSE Command Text with More OPTIONS - Imagine the Possibilities!' +
CAPTION ' ' +
OPTION METER_VISIBLE ON +
|METER_VALUE .vProgress +
|METER_TYPE BAR3D +
|METER_BACK_COLOR WHITE +
|METER_BAR_COLOR NAVY +
|METER_FONT_COLOR RED +
|BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_COLOR GREEN
```

1.17.2.1.4 Gauge Meter

These parameters will display an oscillating meter within the PAUSE window.

Parameter	Value	Description
GAUGE_VISIBLE	ON OFF	Displays an oscillating progress gauge.
GAUGE_COLOR	value	Specifies the color of the progress gauge.
GAUGE_INTERVAL	10-100	Specifies the speed interval of the oscillating progress gauge.

Example:

```
PAUSE 3 USING 'Calculating ... Please Stand By ...' +
CAPTION '          Pause 3 with Gauge ' +
ICON WINDOWS  OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR RED +
|GAUGE_INTERVAL 10 +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR WHITE +
|THEMENAME Steel Blue
```

1.17.3 PLUGIN

Use the PLUGIN command to call upon a file or internal library containing data used to alter, enhance, or extend the operation of R:BASE. R:BASE plugins come in different versions specific to a particular version of R:BASE and are made available through different means. Some plugins are included with the purchase of R:BASE. Other plugins are included with [R:BASE Technologies, Inc. add-on products](#) such as

R:Charts, RBZip, R:PDFWorks, and R:Fax. There is also an exclusive collection of plugins within the [R:BASE Plugin Power Pack](#).

PLUGIN filename parameters

Options

filename

The plugin file name

parameters

Parameters which are used internally by the plugin

Plugins included with R:BASE

- [ExtForm](#)
- [GetDBVersion](#)
- [LoadDirectoryName](#)
- [LoadFileName](#)
- [LoadFileNamePlus](#)
- [R:Calculator](#)
- [R:Choose Builder Plugin](#)
- [R:Dialog Builder Plugin](#)
- [R:Dialog with Buttons Builder Plugin](#)
- [R:MAPI Mail Plugin](#)
- [R:Pause Builder Plugin](#)
- [R:SysDialog Builder Plugin](#)
- [R:Whois Plugin](#)
- [SendKeys](#)
- [SysDialog](#)
- [TimeZone](#)
- [TreeBuilder](#)

Plugins in General

Many software products now used a technology called "PLUGIN". This technology is based on API, which is declared by the program creator. A sample of products which use plugins includes: WinAMP, Adobe Photoshop, ACDSee, etc. It means that the computer programmers can enhance and improve product possibilities using programming languages such as C/C++, Pascal, Basic.

Plugins can be the following type:

- **100** – General type of plugin, which appear in the R:BASE Main Menu under "Utilities" > "Plugins".
- **101** – These plugins are only used with the PLUGIN command, but return a value to R:BASE.

See Also:

Using R:BASE Plugins

1.17.3.1 ExtForm

The ExtForm Plugin is used to capture or assign the version of an external form file.

Syntax:

```
PLUGIN ExtForm VarName|<options>
```

Options:

Option	Value	Description
--------	-------	-------------

ACTION	GET_VERSION_NUMBER SET_VERSION_NUMBER	Specifies to capture or assign the version number of an external form file
FILE	value	Specifies the external form file name
VERSION	value	Specifies the version number for the SET_VERSION_NUMBER parameter

Where:

VarName is the variable name to return "OK" or an error message

vRBTFrmVer is a generated variable for the GET_VERSION_NUMBER parameter, which holds the version number as an INTEGER value

Notes:

If the external form file does not exist, an error is returned to VarName.

Examples:

```
PLUGIN ExtForm vResult|ACTION GET_VERSION_NUMBER|FILE RRBW20.RFF
```

```
SHOW VAR
vRBTFrmVer      = 9          INTEGER
vResult         = OK        TEXT
```

```
PLUGIN ExtForm vResult|ACTION SET_VERSION_NUMBER|FILE TreeView_Sample.RFF|VERSION
10
```

```
SHOW VAR
vResult         = OK        TEXT
```

1.17.3.2 GetDBVersion

The GetDBVersion Plugin is used to capture the version for an R:BASE database.

Syntax:

```
PLUGIN GetDBVersion VarName|<Option>
```

Where:

VarName is the variable to return the database version

Option is the path and structure file name (.RX1, .RB1) for the database

Notes:

The returned variable value will be text.

Examples

```
PLUGIN GetDBVersion vVersion|C:\RBTI\USERMGMT.RX1
```

```
SHOW VAR
vVersion        = V11        TEXT
```

1.17.3.3 LoadDirectoryName

The LoadDirectoryName Plugin is used to load a directory path into a variable by displaying a "Browse for Folder" dialog window for a user to browse the operating system.

Syntax:

```
PLUGIN LoadDirectoryName VarName|<options>
```

Options:

Option	Values	Description
TITLE	value	Specifies the title caption
PATH_NAME	value	Specifies the starting location
WINDOW_STATE	MAX/MIN/NORMAL	Specifies the window state
HEIGHT	value	Specifies the window height
WIDTH	value	Specifies the window width
SHOW_BUTTONS	ON/OFF	Determines if the "Create Folder" and "Delete" buttons are displayed
CONTEXT_MENUS	ON/OFF	Determines if a menu is displayed when user right clicks
READ_ONLY	ON/OFF	Determines if the folders names can be altered
INCLUDE_NON_FOLDERS	ON/OFF	Determines if only folders will be displayed
OLE_DRAG	ON/OFF	Supports drag and drop
OLE_DROP	ON/OFF	Supports drag and drop
CREATE_FOLDER_ICON	ON/OFF	Determines if the "Create Folder" icon is displayed on the button
DELETE_FOLDER_ICON	ON/OFF	Determines if the "Delete" icon is displayed on the button
VIRTUAL_FOLDERS	ON/OFF	Determines if virtual folders can be seen
SHOW_HIDDEN	ON/OFF	Determines if hidden folders can be seen
OK_BUTTON_CAPTION	value	Specifies the caption for the "OK" button
CANCEL_BUTTON_CAPTION	value	Specifies the caption for the "Cancel" button
EXPAND_INITIAL_FOLDER	ON/CASCADE/OFF	Determines if; the initial folder node is expanded (ON), the initial folder and the sub-folders are expanded (CASCADE), the initial folder is not expanded (OFF, default).
AUTO_EXPAND	ON/OFF	Determines if single click will expand/close a node when clicked. Otherwise double click is used. This option is for clicking on nodes, not the tree structure.

Where:

VarName is the variable to return the text value of the selected directory

Notes:

- Each option must be separated by pipe "|" character.
- A value of '[Esc]' will be returned if the [Cancel] button or Close Window [x] button of the folder selection dialog is selected.

Example:

```
PLUGIN LoadDirectoryName vDirName+
|PATH_NAME C:\RBTI\ +
|SHOW_BUTTONS ON+
|CREATE_FOLDER_ICON OFF+
|DELETE_FOLDER_ICON OFF+
```

```
|WIDTH 500+
|HEIGHT 400+
|READ_ONLY ON+
|INCLUDE_NON_FOLDERS OFF+
|SHOW_HIDDEN OFF+
|OK_BUTTON_CAPTION 'Load Folder'+
|EXPAND_INITIAL_FOLDER CASCADE+
|AUTO_EXPAND ON
```

1.17.3.4 LoadFileName

The LoadFileName Plugin is used to load a file name into a variable by displaying a dialog window for the user to browse the operating system.

Syntax:

```
PLUGIN LoadFileName VarName|<options>
```

Options:

Option	Value	Description
FULLPATH	ON/OFF	Determines if the full path for the selected file is returned with the file name
TITLE	value	Specifies the title caption
FILTER	value	Filters the display files (Example: Text Files (*.txt)*.txt)
INITIAL_DIR	value	Specifies the initial directory when launched
NO_CHANGE_DIR	ON/OFF	Determines if the directory changes when the file is selected
MULTISELECT	ON/OFF	Determines if multiple files can be selected
SHOW_HIDDEN	ON/OFF	Determines if hidden files can be seen
NO_LONG_NAMES	ON/OFF	Determines if files not using the 8.3 naming convention are displayed. This is only supported with the OLD_STYLE parameter.
NO_NETWORK_BUTTON	ON/OFF	Determines if the network button is displayed. This is only supported with the OLD_STYLE parameter.
HIDE_READ_ONLY	ON/OFF	Determines if the "Open as Read Only" check box is displayed
NO_DEREFERENCE_LINKS	ON/OFF	Determines if shortcut links are valid or are de-referenced
DONT_ADD_TO_RECENT	ON/OFF	Determines if file selection is added to recently opened documents
ENABLE_SIZING	ON/OFF	Determines if the dialog window can be resized
OLD_STYLE	ON/OFF	Determines if the "old style" window is displayed
NO_VALIDATE	ON/OFF	Determines if validation for invalid characters is used

Where:

VarName is the variable name to return the text value of the selected file

Notes:

- Each option must be separated by pipe "|" character.
- A value of '[Esc]' will be returned if the [Cancel] button or Close Window [x] button of the folder selection dialog is selected.
- Use the pound character for multiple file type filters e.g. **BMP Files (*.bmp)*.bmp#JPEG Files (*.jpg)*.jpg**

Example:

```
PLUGIN LoadFileName vFileName+
```

```
| INITIAL_DIR C:\RBT\ +
| TITLE 'Select file to load' +
| FULLPATH ON+
| NO_CHANGE_DIR ON+
| FILTER JPG Files (*.jpg)#*.jpg+
| HIDE_READ_ONLY ON+
| MULTISELECT ON+
| SHOW_HIDDEN OFF+
| ENABLE_SIZING ON
```

1.17.3.5 LoadFileNamePlus

The LoadFileNamePlus Plugin is used to load a file name into a variable by displaying a dialog window for the user to browse the operating system. The LoadFileNamePlus Plugin was added with additional options to load a file name.

Syntax:

```
PLUGIN LoadFileNamePlus VarName|<options>
```

Options:

Option	Value	Description
FULLPATH	ON/OFF	Determines if the full path for the selected file is returned with the file name
TITLE	value	Specifies the title caption
FILTER	value	Filters the display files (Example: Text Files (*.txt)#*.txt)
INITIAL_DIR	value	Specifies the initial directory when launched
NO_CHANGE_DIR	ON/OFF	Determines if the directory changes when the file is selected
MULTISELECT	ON/OFF	Determines if multiple files can be selected
SHOW_HIDDEN	ON/OFF	Determines if hidden files can be seen
NO_LONG_NAMES	ON/OFF	Determines if files not using the 8.3 naming convention are displayed. This is only supported with the OLD_STYLE parameter.
NO_NETWORK_BUTTON	ON/OFF	Determines if the network button is displayed. This is only supported with the OLD_STYLE parameter.
HIDE_READ_ONLY	ON/OFF	Determines if the "Open as Read Only" check box is displayed
NO_DEREFERENCE_LINKS	ON/OFF	Determines if shortcut links are valid or are de-referenced
DONT_ADD_TO_RECENT	ON/OFF	Determines if file selection is added to recently opened documents
ENABLE_SIZING	ON/OFF	Determines if the dialog window can be resized
OLD_STYLE	ON/OFF	Determines if the "old style" window is displayed
NO_VALIDATE	ON/OFF	Determines if validation for invalid characters is used
VIEW_MODE	THUMBNAILS TILES ICONS (Default) LIST DETAILS	Determines the view mode for the dialog window

Where:

VarName is the variable name to return the text value of the selected file

Notes:

- Each option must be separated by pipe "|" character.

- A value of '[Esc]' will be returned if the [Cancel] button or Close Window [x] button of the folder selection dialog is selected.
- Use the pound character for multiple file type filters e.g. **BMP Files (*.bmp)*.bmp#JPEG Files (*.jpg)*.jpg**

Example:

```
PLUGIN LoadFileNamePlus vFileName+
| INITIAL_DIR C:\RBTI\ +
| TITLE 'Select file' +
| FULLPATH ON+
| NO_CHANGE_DIR ON+
| FILTER PDF Files (*.pdf)*.pdf+
| HIDE_READ_ONLY ON+
| MULTISELECT ON+
| SHOW_HIDDEN OFF+
| ENABLE_SIZING ON+
| VIEW_MODE LIST
```

1.17.3.6 R:Calculator

The R:Calculator Plugin is used to launch a calculator to perform any of the standard operations for which you would normally use a hand-held calculator. An optional theme can be added to the calculator window.

Syntax:

```
PLUGIN RCalculator vResult|THEMENAME Themename
```

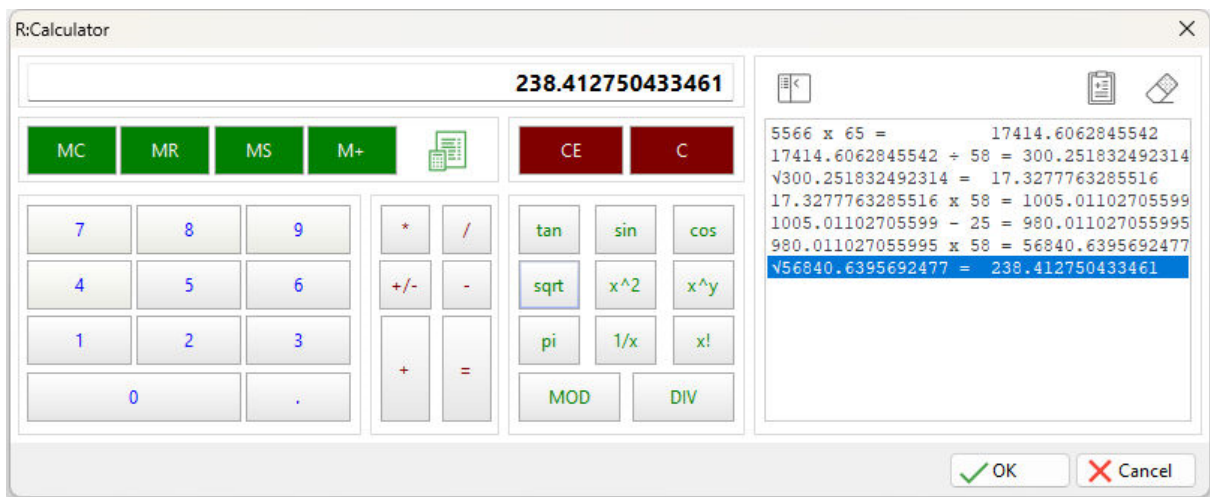
Where:

vResult is the variable to return the value of the calculated result

Themename will display the calculator using a specified theme

Notes:

- The returned variable value will be the correct data type of the calculated result (INTEGER, DOUBLE, REAL).
- If no variable name is specified, the variable vRBTICalcValue will return the result.



1.17.3.7 R:Choose Builder Plugin

The R:Choose Builder Plugin is a syntax builder for the [CHOOSE](#) command. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

Using this plugin you can easily create choose windows to prompt end users for information within your R:BASE application.

R:Choose Builder Plugin

General Information

Options

Ext Options

Syntax

CHOOSE type

#COLUMNS in Table/View

Tables/Views

Table/View Name ^

SalesDataByCompany

SalesInf

SecurityTable

Staff

StaffByDept

StateAbr

SYS_COLUMNS

SYS_COMMENTS

SYS_CONSTRAINTS

SYS_DEFAULTS

SYS_FORMS

SYS_FORMS2

Check

☒ None

☐ CHKBX

☐ CHKSORT

Choices: 0

Lines: 5

☐ Formatted

Caption/Title

Caption: CHOOSE Caption Here...

Title: CHOOSE TITLE Here...

Variable

vChoose

Test

Build

Close

R:Choose Builder Plugin

General Information Options Ext Options Syntax

☒ List Options
Font Color: AaBbYyZz Background Color: Button Face

☒ Title Options
Font Color: AaBbYyZz Background Color: Button Face

☒ Window
☐ No Caption
☐ Small Caption
☒ Normal Caption
☐ Show "Select All" Button Background Color: Button Face
☐ Single Click

☒ Transparency
255

☒ Buttons
OK Caption: OK Back Color: Default
Cancel Caption: Cancel Button Color: Default
Font Color: AaBbYyZz Size (W x H): 75 x 75
☐ Show Glyph

☒ Other
Left: 10 Top: 50
Theme Name:

☐ Initial Values
List Position: Add... Edit... Delete Clear
List Content: Add... Edit... Delete Clear

Test Build Close

R:Choose Builder Plugin

General Information Options Ext Options Syntax

Grid Options

☒ Zebra Stripe

Odd Color: ☐ 3D Light

Even Color: ☐ Window

☒ Show Horizontal Lines

☒ Show Vertical Lines

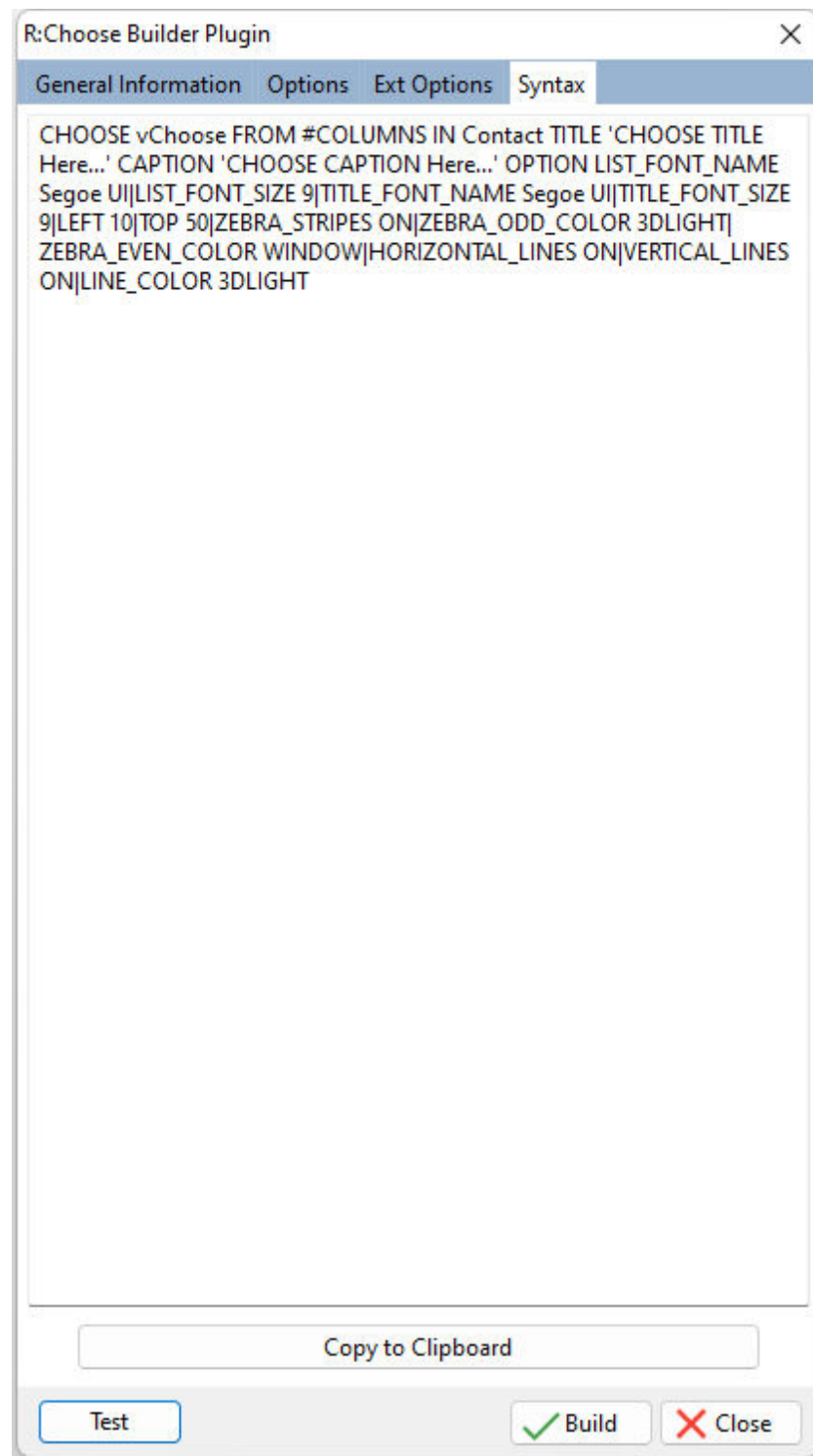
Line Color Color: ☐ 3D Light

Vertical Line Breaks:

Headers:

Text Breaks:

Test Build Close



1.17.3.8 R:Dialog Builder Plugin

The R:Dialog Builder Plugin is a syntax builder for the [DIALOG](#) command. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

Using this plugin you can easily create dialog windows to prompt end users for information within your R:BASE application.

The screenshot shows the 'R:Dialog Builder Plugin' window with three tabs: 'General Information', 'Options', and 'Syntax'. The 'General Information' tab is active. It contains the following sections:

- Message Text:** A text input field with the placeholder 'Type Message Here ...'. Below it is a checked checkbox labeled 'Show Accelerator Character'.
- Variables:** Two input fields: 'Response:' with the value 'vResponse' and 'End Key:' with the value 'vEndKey'.
- Format:** A radio button labeled 'Lines' is selected, followed by a text input field containing the number '1'. A radio button labeled 'Password' is unselected. Below this is an unchecked checkbox labeled 'Show ROSK Button'.
- Caption:** A text input field with the placeholder 'Type Caption Here ...'.
- DIALOG Icons:** A grid of 12 icons, each with a radio button. The 'Information' icon (a blue circle with a white 'i') is selected. The other icons are: Application (gears), Help (globe), Stop (red octagon), Attention (triangle with exclamation mark), Confirmation (green checkmark), Question (circle with question mark), Warning (yellow triangle with exclamation mark), Error (red circle with X), Serious (blue triangle with exclamation mark), and Windows (blue square with four smaller squares).

At the bottom of the window are three buttons: 'Test', 'Build' (with a green checkmark icon), and 'Close' (with a red X icon).

R:Dialog Builder Plugin

General Information Options Syntax

☒ Message

Font Color: ■ Default

Background Color: □ Button Face

Font Name: ■ MS Sans Serif Size: 10

☐ Bold ☐ Italic ☐ Underline ☐ Strikeout

☒ Buttons

OK Caption: &OK Cancel Caption: &Cancel

OK Button Color: □ Defau Cancel Button Color: □ Defaul

OK Button Width: 75 Cancel Button Width: 75

OK Button Font: OK Cancel Button Font: Cancel

Background Color: □ Buttoi ☐ Show Glyph

Button Height: 25

☒ Window

☐ No Caption ☐ Small Caption ☒ Normal Caption

Background Color: □ Button Face

☐ Transparency

255

Icon File Name

☐ Use ■

☐ Resize Icon

☒ Other

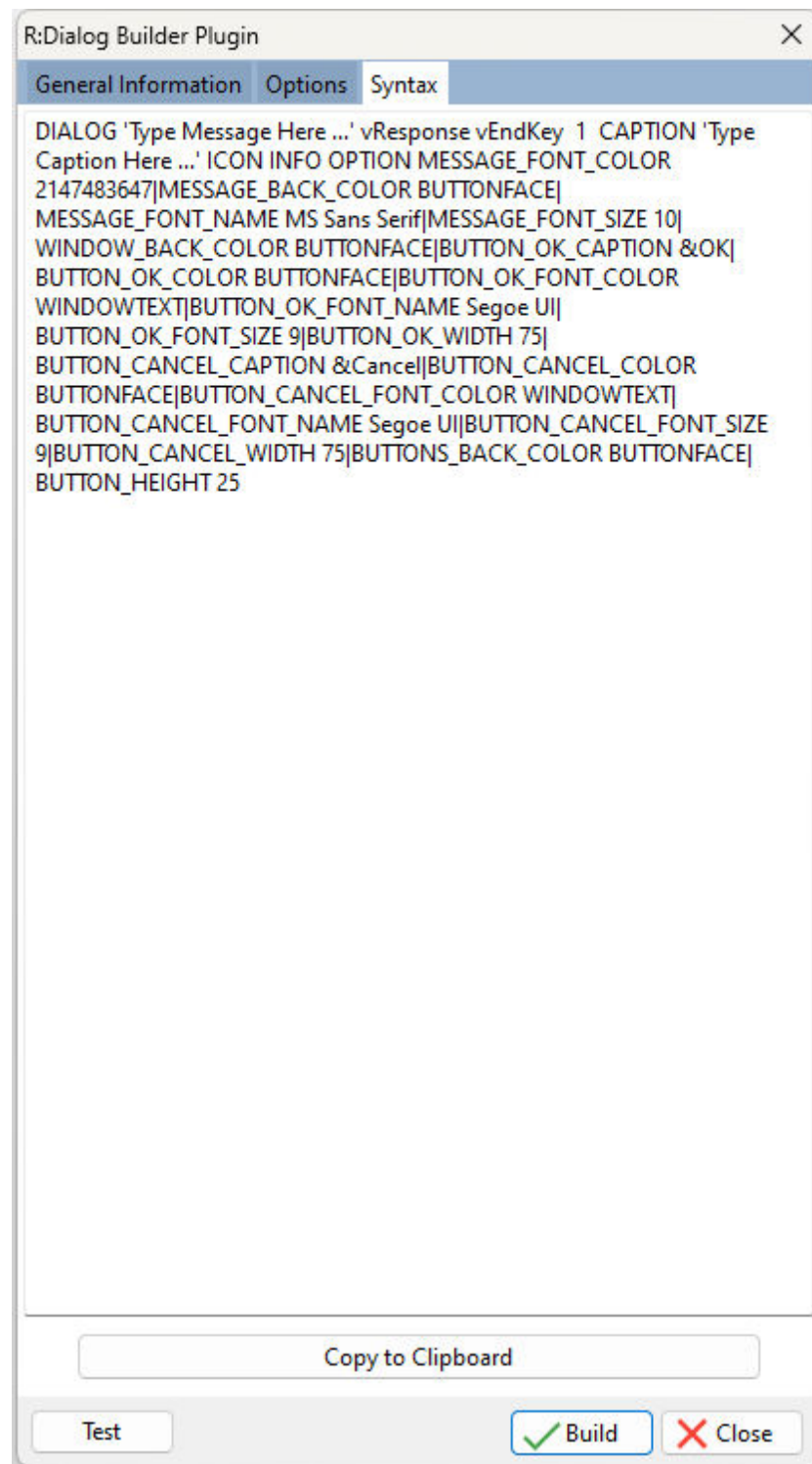
Left: -1 Top: -1 Right: -1 Bottom: -1

Theme Name: ■

☐ Use Modern Look

☐ Disable Generation of Custom Formats

Test Build Close



1.17.3.9 R:Dialog with Buttons Builder Plugin

The R:Dialog with Buttons Builder Plugin is a syntax builder for the [DIALOG](#) command. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

Using this plugin you can easily create dialog windows with buttons for end users to select within your R:BASE application.

The screenshot shows a dialog box titled "R:Dialog with Buttons Builder Plugin" with a close button (X) in the top right corner. The dialog has three tabs: "General Information", "Options", and "Syntax". The "General Information" tab is selected. It contains the following sections:

- Message Text:** A text input field with the placeholder "Type Message Here ...". Below it is a checked checkbox labeled "Show Accelerator Character".
- Variables:** Two text input fields. The first is labeled "Response:" and contains "vResponse". The second is labeled "End Key:" and contains "vEndKey".
- Default Button:** Two radio buttons. The first is labeled "Yes" and is selected. The second is labeled "No".
- Caption:** A text input field with the placeholder "Type Caption Here ...".
- DIALOG Icons:** A grid of 12 icons, each with a radio button next to it:
 - Application (gears icon)
 - Attention (warning triangle icon)
 - Confirmation (checkmark icon)
 - Error (red X icon)
 - Help (globe icon)
 - Information (blue i icon) - This is selected.
 - Question (question mark icon)
 - Serious (blue exclamation triangle icon)
 - Stop (red octagon icon)
 - Warning (yellow exclamation triangle icon)
 - Windows (blue square icon)

At the bottom of the dialog, there are three buttons: "Test", "Build" (with a green checkmark icon), and "Close" (with a red X icon).

R:Dialog with Buttons Builder Plugin

General Information Options Syntax

☒ Message

Font Color: ■ Default

Background Color: □ Button Face

Font Name: 🔍 Segoe UI Size 9

☐ Bold ☐ Italic ☐ Underline ☐ Strikeout

☒ Window

☐ No Caption ☐ Small Caption ☒ Normal Caption

Background Color: □ Button Face

☒ Buttons

Yes Caption: &Yes No Caption: &No

Yes Button Color: □ Default No Button Color: □ Default

Yes Button Width: 75 No Button Width: 75

Yes Button Font: Yes No Button Font: No

Button Height: 25

☐ Transparency

255

Icon File Name

☐ Use 📁

☐ Resize Icon

☒ Other

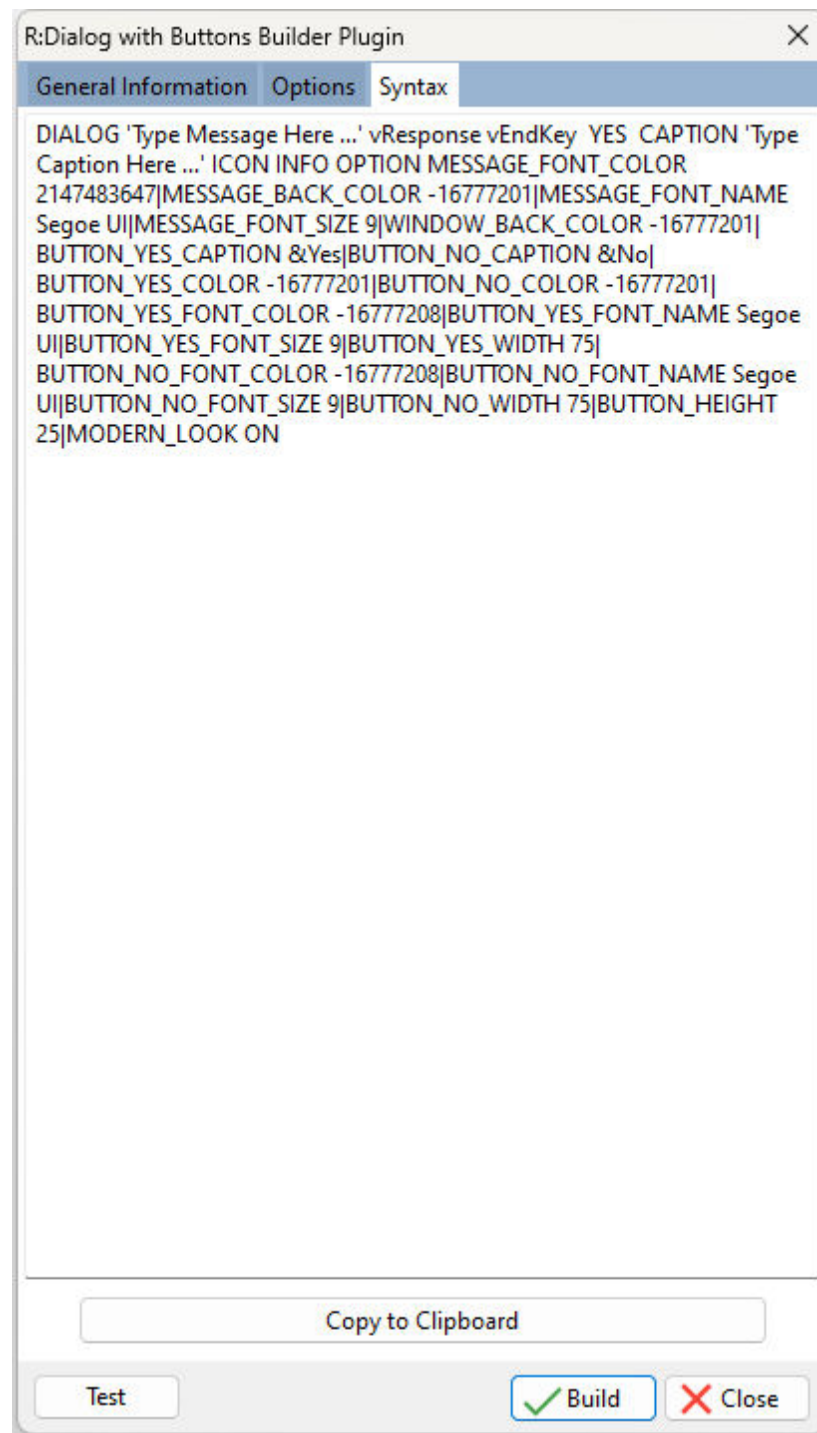
Left: -1 Top: -1 Right: -1 Bottom: -1

Theme Name:

☒ Use Modern Look

☐ Disable Generation of Custom Formats

Test Build Close



1.17.3.10 R:MAPI Mail Plugin

The R:MAPI Mail Plugin is a utility to send a message through your email client program. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

The screenshot shows a Windows-style dialog box titled "R:MAPI Mail Plugin". It contains several sections: "Account Settings" with fields for "Account Name:" and "Password:"; "Address Settings" with fields for "To:", "Cc:", "Bcc:", and "Subject:", each with a small up/down arrow on the right; a large "Message" text area; and an "Attachments" section with a list box. Below the list box are two buttons: "+ Attach File(s)" and "- Remove Attachment". At the bottom, there are two checkboxes: "Send Directly" and "Resolve Names", followed by "Send" and "Cancel" buttons with checkmark and X icons respectively.

1.17.3.11 R:Pause Builder Plugin

The R:Pause Builder Plugin is a syntax builder for the [PAUSE](#) command. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

Using this plugin you can easily create message windows to display information within your R:BASE application.

R:Pause Builder Plugin

General Information Options More Options Syntax

PAUSE Type

☐ 1 ☒ 2 ☐ 3 ☐ 4












Message Text

Message Text

Caption

Caption Text

PAUSE Icons

 <input type="radio"/> Application	 <input type="radio"/> Help	 <input type="radio"/> Stop
 <input type="radio"/> Attention	 <input checked="" type="radio"/> Information	 <input type="radio"/> Warning
 <input type="radio"/> Confirmation	 <input type="radio"/> Question	 <input type="radio"/> Windows
 <input type="radio"/> Error	 <input type="radio"/> Serious	

Button Text

Button Text

Test Build Close

R:Pause Builder Plugin

General Information Options More Options Syntax

Icon File Name
☐ Use

Pause Window Background Color
☒ Use

Message Color
☒ Use

☒ Message Font
Name
Color Size
☐ Bold ☐ Italic ☐ Underline ☐ Strikeout

Button Color
☒ Use
☐ Width ☐ Height

☒ Button Font
Name
Color Size
☐ Bold ☐ Italic ☐ Underline ☐ Strikeout

Transparency
☐ Use

☒ Other
Left: Top:
Theme Name:

Test Build Close

R:Pause Builder Plugin

General Information Options More Options Syntax

☒ Gauge

Color ■ Default Interval -1

☒ Meter

Type Normal

Value -1

Maximum Value -1

Font Color ■ Default

Back Color ■ Default

Bar Color ■ Default

Off Color ■ Default

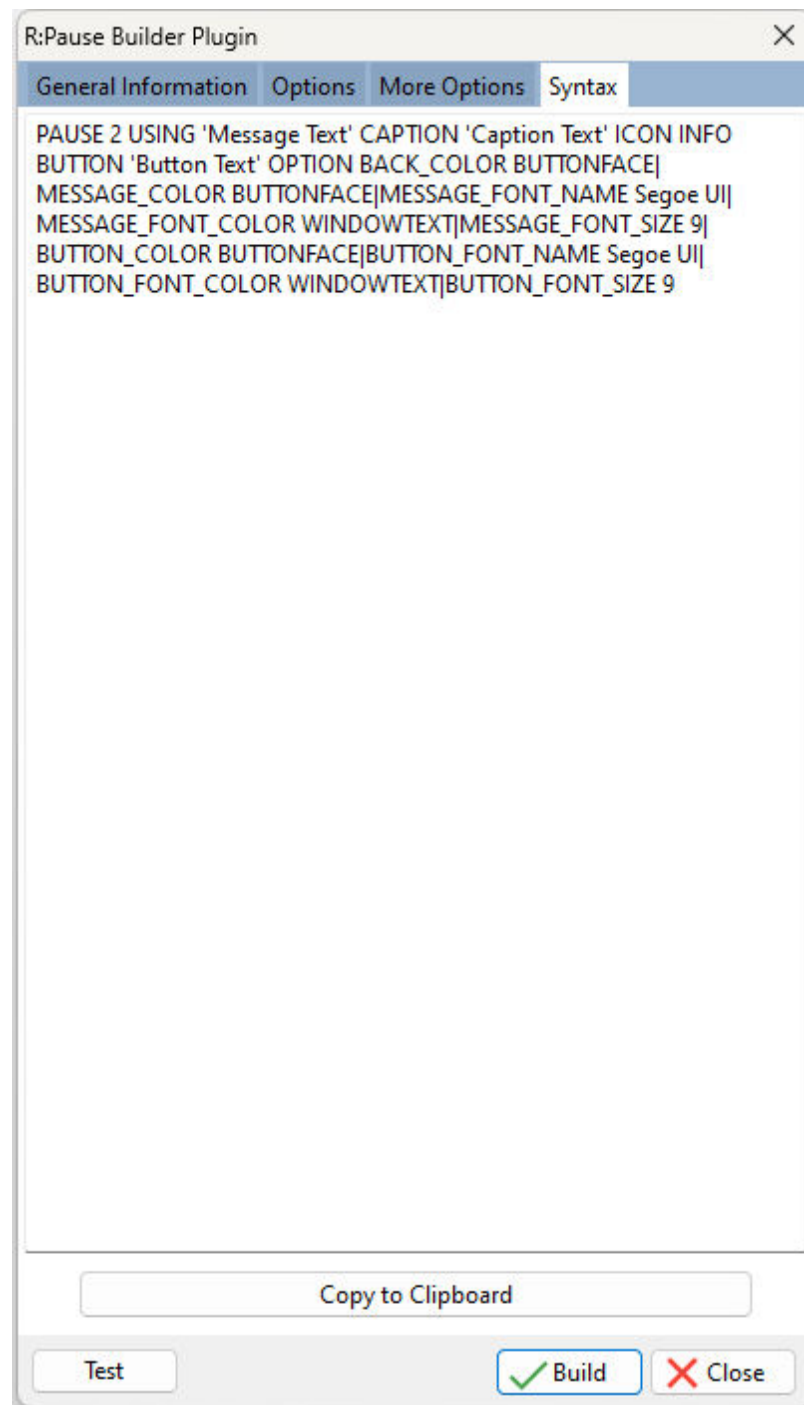
Color 1 ■ Default

Color 2 ■ Default

Color 3 ■ Default

☐ No Focus

Test ✓ Build ✗ Close



1.17.3.12 R:SysDialog Builder Plugin

The R:SysDialog Builder Plugin is a syntax builder for the creating system dialogs. It appears in the R:BASE for Windows Menu Bar under "Utilities" > "Plugins". The [SysDialog](#) syntax can be modified as needed.

Using the plugin dialog windows can be easily created with several buttons, for end users to select within your R:BASE application. The dialog can display several provided icon images, and a default button can

be assigned. The dialog window displays as per the native operating system environment and allows no customized display features.

R:SysDialog Builder Plugin

General Information | **Syntax**

Return Variable
vResponse

Caption
Message

Message
Please confirm to proceed!

Icon

☒ Information ☒ Confirmation
☐ Warning ☐ Error

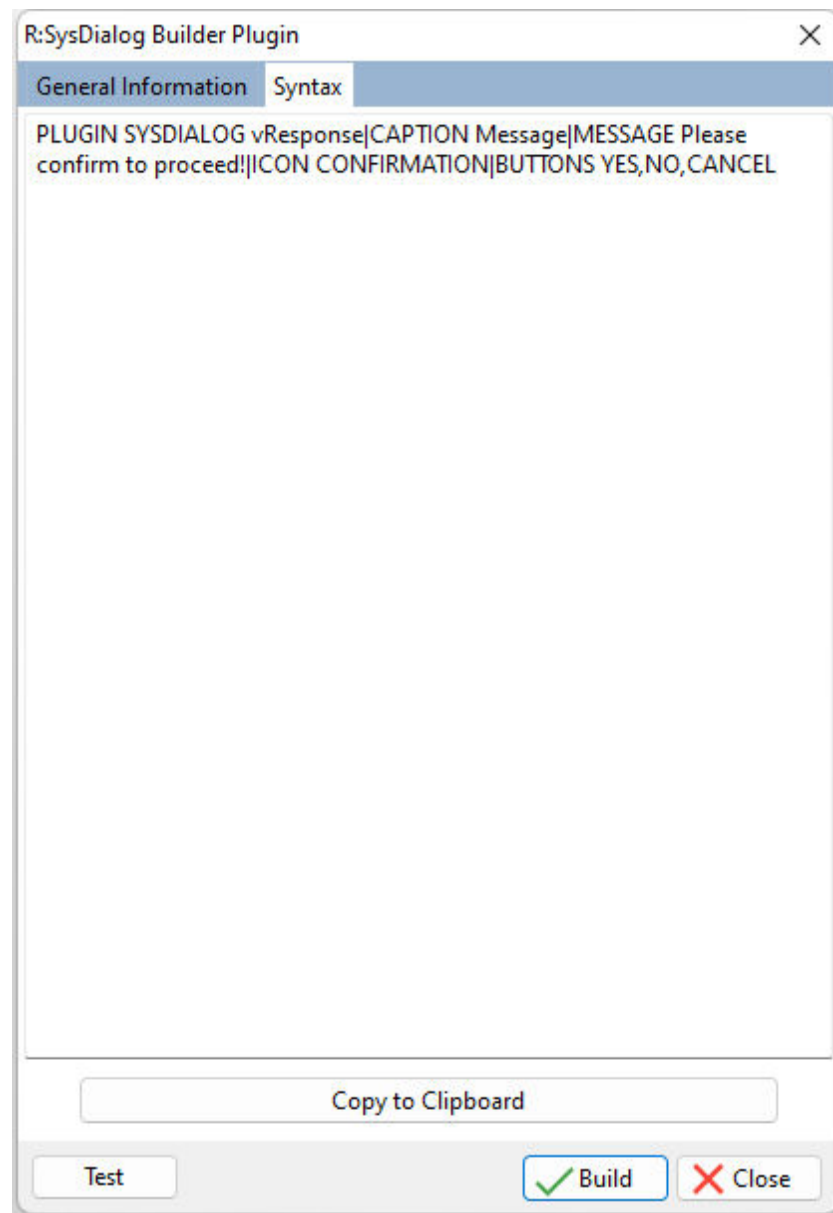
Buttons

☒ YES ☐ RETRY
☒ NO ☐ IGNORE
☐ OK ☐ ALL
☒ CANCEL ☐ NO TO ALL
☐ ABORT ☐ YES TO ALL

Default Button

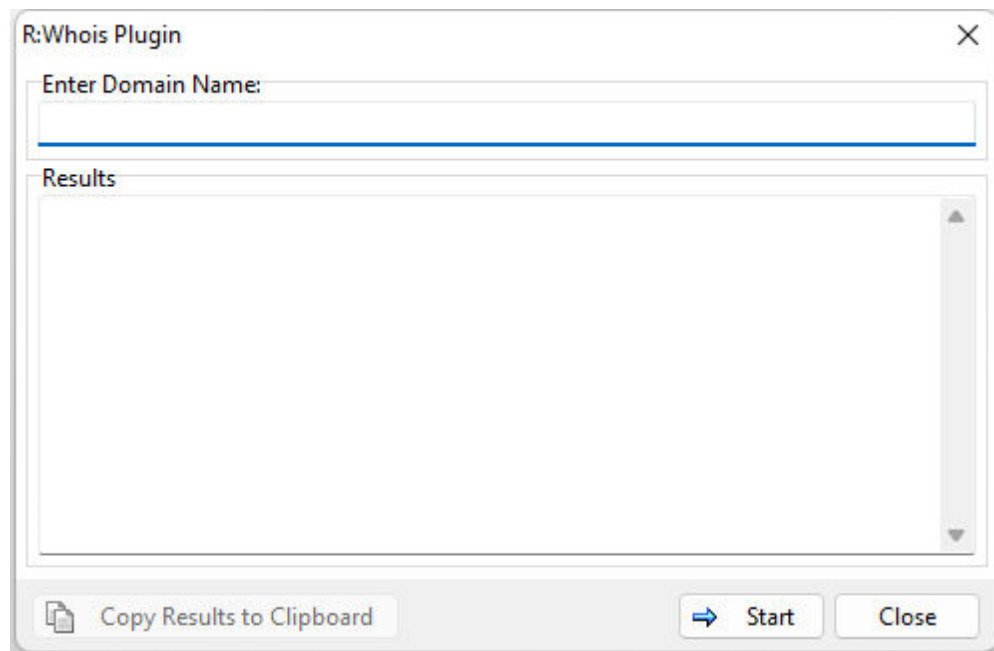
☒ YES ☐ RETRY
☐ NO ☐ IGNORE
☐ OK ☐ ALL
☐ CANCEL ☐ NO TO ALL
☐ ABORT ☐ YES TO ALL

Test **Build** **Close**



1.17.3.13 R:Whois Plugin

The R:Whois Plugin will find and display domain name registration information and additional Web site data. The Plugin appears in the R:BASE main menu bar under "Utilities" > "Plugins".



1.17.3.14 SendKeys

The SendKeys plugin sends one or more keystrokes to the active control, as if typed at the keyboard.

Syntax:

```
PLUGIN SendKeys vResult|<Options>
```

Options:

Option	Value	Description
KEYS	value	Specifies the keys to be simulated, or code for special keys that aren't displayed when you press a key
WINDOW_NAME	value	Specifies the text that appears in the target window's title bar. The value can be left empty.
WAIT	ON OFF (Default)	Specifies the wait for the keys to be processed by the recipient before returning control

Where:

vResult is the variable name to return "OK" or an error message

Notes:

- Each option must be separated by the pipe "|" character.
- It is advised to be very careful with the SendKeys plugin capability. If the sequence of commands gets interrupted by another process, and the focus is changed, unpredictable results may occur.
- Special keys are enclosed in braces. As braces are used to enclose [comments](#) in R:BASE, quote the text so it is not interpreted as a comment.

Special Keys

Key	Code
-----	------

BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
DEL or DELETE	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER	{ENTER} or ~
ESC	{ESC}
HELP	{HELP}
HOME	{HOME}
INS or INSERT	{INSERT} or {INS}
LEFT ARROW	{LEFT}
NUM LOCK	{NUMLOCK}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
PRINT SCREEN	{PRTSC}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
TAB	{TAB}
UP ARROW	{UP}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}

To specify keys combined with any combination of the SHIFT, CTRL, and ALT keys, precede the key code with one or more of the following codes:

Key	Code
SHIFT	+
CTRL	^
ALT	%

Examples:

Example 01.

```
SET VAR vResult TEXT = NULL
```

--Set focus to an R:BASE Variable Memo control, then wait for 1 second before sending keys. Note the Variable Memo is set in the properties to accept the [Enter] key.

```
PROPERTY VariableMemo1 SET_FOCUS ' '
```

```
PAUSE FOR 1
```

```
PLUGIN SENDKEYS vResult|KEYS This text is displayed using the SendKeys plugin.~~
```

```
PLUGIN SENDKEYS vResult|KEYS 'This is one of the many "key" features in R:BASE!'
```

--The Ctrl+Home moves to the beginning of the content.

```
PLUGIN SENDKEYS vResult|KEYS '^ {HOME}'
```

--Adds text and then Enter

```
PLUGIN SENDKEYS vResult|KEYS Greetings from R:BASE!~~
```

--Ctrl+End moves to the end of the content then presses Enter

```
PLUGIN SENDKEYS vResult|KEYS '^{\END}~~'
```

```
PLUGIN SENDKEYS vResult|KEYS Thank you for choosing R:BASE!
```

--Highlights the last line. this is Home, Shift+End

```
PLUGIN SENDKEYS vResult|KEYS '{HOME}+{\END}'
```

Example 02.

```
SET VAR vResult TEXT = NULL
```

--Launch notepad and wait for 1 second to make sure notepad is loaded

```
LAUNCH NOTEPAD.EXE
```

```
PAUSE FOR 1
```

--While WINDOW_NAME is only required once to activate the window, it is safe to always specify it.

```
PLUGIN SENDKEYS vResult|WINDOW_NAME Untitled - Notepad|KEYS This text is displayed  
using the SendKeys plugin.~~
```

```
PLUGIN SENDKEYS vResult|WINDOW_NAME Untitled - Notepad|KEYS 'This is one of the  
many "key" features in R:BASE!'
```

1.17.3.15 SysDialog

The SysDialog plugin creates system dialogs to capture a response from end users. The dialog window supports several buttons to select an option. The dialog can display several provided icon images, and a default button can be assigned. The dialog window displays as per the native operating system environment and allows no customized display features. A syntax builder for the plugin is available within the R:BASE for Windows Menu Bar under "Utilities" > "Plugins".

Syntax:

```
PLUGIN SysDialog vResponse|<Options>
```

Options:

Option	Value	Description
MESSAGE	value	Specifies the dialog message text
CAPTION	value	Specifies the caption bar text
ICON	WARNING ERROR INFORMATION CONFIRMATION	Specifies the icon to be displayed. The default icon is INFORMATION.
BUTTONS	YES NO OK CANCEL ABORT RETRY IGNORE ALL NO_TO_ALL YES_TO_ALL	Specifies the buttons to display. Multiple buttons can be added by using a comma.
DEFAULT_BUTTON	YES NO OK CANCEL ABORT RETRY IGNORE	Specifies the default button to have focus

	ALL NO_TO_ALL YES_TO_ALL	
--	--------------------------------	--

Where:

vResponse is the variable name to return the selected button (OK, CANCEL, ABORT, RETRY, IGNORE, YES, NO, ALL, NO_TO_ALL, YES_TO_ALL)

Notes:

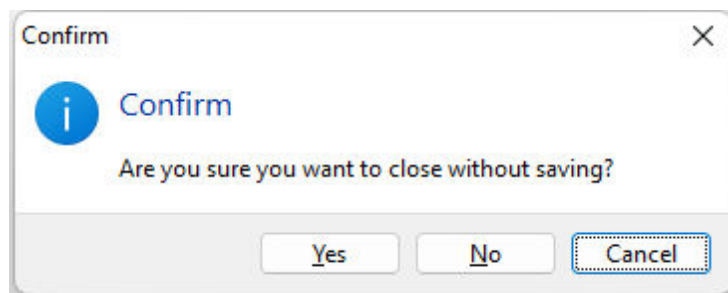
- Each option must be separated by pipe "|" character.
- The button sequence appear in the following order: Yes, No, OK, Cancel, Abort, Retry, Ignore, All, No_To_All, Yes_To_All
- Common button combinations include "ABORT,RETRY,IGNORE", "CANCEL,RETRY,IGNORE", "OK, "OK,CANCEL", "CANCEL,RETRY", "YES,NO", and "YES,NO,CANCEL"

Rules:

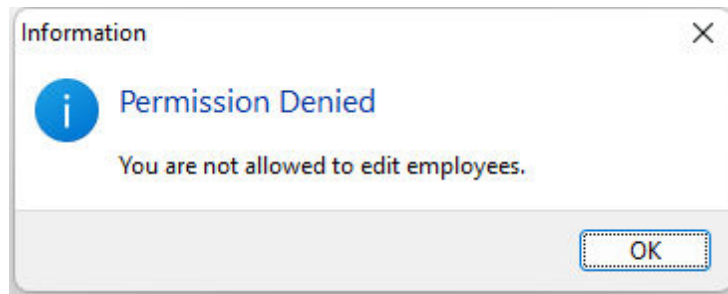
1. The standard modal form with buttons has a DEFAULT button and a CANCEL button.
2. The "Default button" is the button that gets clicked if [Enter] is pressed and there's no focused control. The order of detection is: OK (if visible), YES (if visible), RETRY (if visible), first visible button
3. The "Cancel button" is the button that gets clicked if [Esc] s pressed. The order of detection is: CANCEL (if visible), NO (if visible), OK (if visible)
4. If CANCEL, NO, or OK are not in the list of visible buttons, then pressing [Esc] will return CANCEL
5. If the [X] button is clicked the result is CANCEL

Examples:**Example 1:**

```
PLUGIN SYSDIALOG vResponse+  
|MESSAGE Are you sure you want to close without saving?+  
|CAPTION Confirm+  
|ICON CONFIRMATION+  
|BUTTONS YES,NO,CANCEL+  
|DEFAULT_BUTTON CANCEL
```

**Example 2:**

```
PLUGIN SYSDIALOG vResponse+  
|MESSAGE You are not allowed to edit employees.+  
|CAPTION Permission Denied+  
|ICON INFORMATION+  
|BUTTONS OK
```

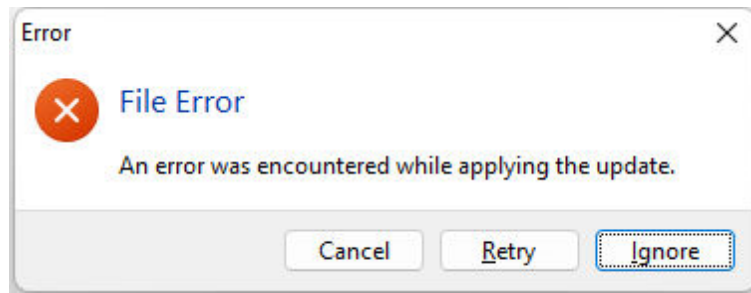


Example 3:

```

PLUGIN SYSDIALOG vResponse+
|MESSAGE An error was encountered while applying the update.+
|CAPTION File Error+
|ICON ERROR+
|BUTTONS CANCEL,RETRY,IGNORE+
|DEFAULT_BUTTON IGNORE

```



1.17.3.16 TimeZone

The TimeZone plugin retrieves the current time zone and daylight saving time settings for the computer. If the computer is configured for daylight saving time, the plugin returns date and local time when the transition from standard time to daylight saving time, and vice versa, occurs on the operating system.

Syntax:

```

PLUGIN TimeZone vResult|ACTION GET_INFO

```

Where:

The following return variables are generated:

Variable	Description
vResult	<p>Returns the current daylight saving time setting:</p> <p>OK - DST_IN_DAYLIGHT_DATE - daylight saving time auto adjustment is enabled and current time is in daylight date</p> <p>OK - DST_IN_STANDARD_DATE - daylight saving time auto adjustment is enabled and current time is in standard date</p> <p>OK - DST_NOT_USED - daylight saving time is not supported, or the automatic daylight saving time adjustment is disabled</p> <p>-ERROR- TIME_ZONE_ID_INVALID - an error occurred</p>
vrBTITZBias	Returns the current bias for local time translation on the computer, in minutes. The bias is the difference, in minutes, between Coordinated

	Universal Time (UTC) and local time. All translations between UTC and local time are based on the following formula: UTC = local time + bias
vRBTITZStdName	Returns the standard time zone name
vRBTITZStdDate	Returns the date and local time when the transition from daylight saving time to standard time occurs on the operating system
vRBTITZStdBias	Returns the bias value to be used during local time translations that occur during standard time. This value is added to vRBTITZBias to form the bias used during standard time. In most time zones, the value is zero.
vRBTITZDLName	Returns the daylight saving time name
vRBTITZDLDate	Returns date and local time when the transition from standard time to daylight saving time occurs on the operating system
vRBTITZDLBias	Returns the bias value to be used during local time translations that occur during daylight saving time. This value is added to vRBTITZBias to form the bias used during daylight saving time. In most time zones, the value is -60.

Examples:**Example 01:**

The below variables are generated for a computer in the Eastern Standard Time zone, automatically adjusted to ON for daylight saving time:

Variable	= Value	Type
-----	-----	-----
vRBTITZBias	= 300	INTEGER
vRBTITZStdName	= Eastern Standard Time	TEXT
vRBTITZStdDate	= 11/01/0000 02:00:00:00	TEXT
vRBTITZStdBias	= 0	INTEGER
vRBTITZDLName	= Eastern Daylight Time	TEXT
vRBTITZDLDate	= 03/02/0000 02:00:00:00	TEXT
vRBTITZDLBias	= -60	INTEGER
vResult	= OK - DST_IN_DAYLIGHT_DATE	TEXT

Example 02:

The below variables are generated for a computer in the Pacific Standard Time zone, with the automatic daylight saving time adjustment setting disabled:

Variable	= Value	Type
-----	-----	-----
vRBTITZBias	= 480	INTEGER
vRBTITZStdName	= Pacific Standard Time	TEXT
vRBTITZStdDate	= 00/00/0000 00:00:00:00	TEXT
vRBTITZStdBias	= 0	INTEGER
vRBTITZDLName	= Pacific Standard Time	TEXT
vRBTITZDLDate	= 00/00/0000 00:00:00:00	TEXT
vRBTITZDLBias	= 0	INTEGER
vResult	= OK - DST_NOT_USED	TEXT

1.17.3.17 TreeBuilder

The TreeBuilder Plugin is available to create a "tree table", with all necessary information to build a tree-like presentation in reports.

Syntax:

```

PLUGIN TreeBuilder vResult +
|TABLE <TableName> +
|ID <IDColumn> +
|PARENT <ParentColumn> +

```

```
|ORDER <OrderColumn> +
|TREE_TABLE <TreeTableName>
```

Options:

Option	Value	Description
TABLE	value	Specifies the source table
ID	value	Specifies the ID field in the source table
PARENT	value	Specifies the parent field in the source table
ORDER	value	Specifies the field that controls sorting (optional). This is usually the field to be the node's text.
TREE_TABLE	value	Specifies the newly created output table

Where:

vResult is the text variable to return the status, such as "OK" or the exact -ERROR- message

Notes:

The created "Tree Table" output table has the following structure:

```
CREATE TABLE <TreeTable> +
(TREE_TABLE_ID INTEGER, +
TREE_TABLE_PARENT_ID INTEGER, +
TREE_TABLE_ROW_ID INTEGER, +
TREE_LEVEL INTEGER, +
TREE_HAS_CHILDREN INTEGER)
```

- TREE_TABLE_ID - Incrementing integer starting with 1
- TREE_TABLE_PARENT_ID - Value of the source table's parent field
- TREE_TABLE_ROW_ID - Value of source table's ID field
- TREE_LEVEL - The node's level. Outermost node is zero (0), then next level is one (1), ...
- TREE_HAS_CHILDREN - Indicator if node has children. One (1) if node has children, zero (0) if no child nodes

RRBYW20 Sample:

```
PLUGIN TreeBuilder vResult +
|TABLE Departments +
|ID DepartmentID +
|PARENT OwnerDept +
|ORDER DepartmentID +
|TREE_TABLE DeptTree
```

A view is created based on the tree output table and the source table. The view is then used as the report's data source.

```
CREATE VIEW `DepartmentTree` +
(TREE_TABLE_ID,TREE_TABLE_PARENT_ID,TREE_TABLE_ROW_ID,+
TREE_LEVEL,TREE_HAS_CHILDREN,+
TREE_VIEW_CAPTION,DepartmentID,Description,OwnerDept) AS +
SELECT T1.TREE_TABLE_ID,T1.TREE_TABLE_PARENT_ID, +
T1.TREE_TABLE_ROW_ID,T1.TREE_LEVEL,T1.TREE_HAS_CHILDREN,+
((SFIL(' ', T1.TREE_LEVEL*5)) + T2.Description), +
T2.DepartmentID,T2.Description,T2.OwnerDept +
FROM DeptTree T1 LEFT OUTER JOIN Departments T2 +
ON T1.TREE_TABLE_ROW_ID = T2.DepartmentID
```

With the following variables defined in the report expression builder;

```
1 : INTEGER : D : vTreeLevel = TREE_LEVEL
```

```
2 : INTEGER : D : vHasChildren = TREE_HAS_CHILDREN
```

the report Detail band's "On Before Generate" EEP can use the TREE_LEVEL and TREE_HAS_CHILDREN fields to manipulate the label and image locations, to emulate a tree view presentation.

```
-- Use TREE_LEVEL and TREE_HAS_PARENT to indent for the tree view
```

```
--Node label
SET VAR vMargin DOUBLE = 1.5
SET VAR vLevelIndent DOUBLE = 0.2
SET VAR vLeft = .vMargin + (.vLevelIndent * .vTreeLevel)
PROPERTY lblCaption LEFT .vLeft
```

```
--Node image. A few mms to the left of the label
SET VAR vLeft = (.vLeft - 0.3)
PROPERTY imgParent LEFT .vleft
PROPERTY imgChild LEFT .vleft
IF vHasChildren = 0 THEN
    PROPERTY imgParent VISIBLE FALSE
    PROPERTY imgChild VISIBLE TRUE
ELSE
    PROPERTY imgParent VISIBLE TRUE
    PROPERTY imgChild VISIBLE FALSE
ENDIF
RETURN
```

Using this approach, a tree view presentation with unlimited node levels can be created. In addition, the tree structure in a report can be created with very complex appearance (e.g. a sub report inside each node). This was made possible by the TREE_LEVEL and TREE_HAS_CHILDREN fields.

1.17.4 PREVROW

Use the PREVROW command in an entry/exit procedure (EEP) to move to the previous row.

PREVROW

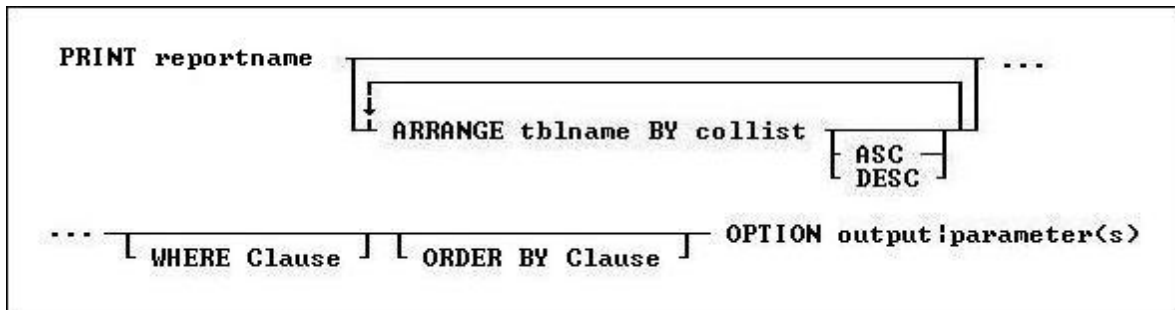
1.17.5 PREVTAB

Use the PREVTAB command in an entry/exit procedure (EEP) to move to the previous table.

PREVTAB

1.17.6 PRINT

Use the PRINT command to send a report to the screen, printer, or file.



Options

rptname

Specifies the name of the report you want to print.

ARRANGE tblname BY collist

Sorts the rows displayed by a report's lower-level tables. Each lower-level table in the report that you want to sort requires its own ARRANGE clause. Now you can specify up to twenty-five tables to ARRANGE and up to five columns in each table on which R:BASE will sort.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

OPTION

Specifies output options of the report along with additional parameters for each output such as:

- [SCREEN](#)
- [PRINTER](#)
- [BMP](#)
- [DIF](#)
- [EMF](#)
- [ETXT](#) (Report Emulation Text)
- [GIF](#)
- [HTML](#)
- [JPG](#)
- [LOTUS](#)
- [PDF](#)
- [PNG](#)
- [QPRO](#)
- [RTF](#)
- [SVG](#)
- [SYLK](#)
- [TIFF](#)
- [TXT](#)
- [WMF](#)
- [XHTML](#)
- [XLS](#)
- [XLS2003](#)
- [XLSX](#)

Working with Breakpoints

When a report has breakpoints, R:BASE automatically sorts by the defined breakpoint columns. You can sort additional columns other than breakpoint columns if the breakpoint columns are listed in the [ORDER BY](#) clause first. If an ORDER BY clause that overrides the breakpoint sort is included in the PRINT command, the report might not produce the results defined by the breakpoints.

1.17.6.1 SCREEN

The SCREEN output option displays the report in a Print Preview window.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SCREEN
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
WINDOW_STATE	MAXIMIZED MINIMIZED NORMAL	Specifies the state of preview window. By default, the window state is normalized.
TOOLBAR_VISIBLE	TRUE FALSE	Specifies the option to display preview toolbar with all buttons such as, Print, Whole Page, Page Width, 100%, Zoom Level, First, Page Previous Page, Next Page, Last Page and Cancel/Close.
BUTTONBAR_VISIBLE	TRUE FALSE	Specifies the option to display button bar with Whole Page, Page Width and 100%. If set to FALSE, all three buttons (Whole Page, Page Width and 100%) will not be visible.
BTNFIND_VISIBLE	TRUE FALSE	Specifies the option to display the Find Text button on the toolbar
BTNOUTLINE_VISIBLE	TRUE FALSE	Specifies the option to display the Report Outline button on the toolbar
BTNPRINT_VISIBLE	TRUE FALSE	Specifies the option to display Printer button on the toolbar
BTNWHOLEPAGE_VISIBLE	TRUE FALSE	Specifies the option to display Whole Page button on the toolbar.
BTNPAGEWIDTH_VISIBLE	TRUE FALSE	Specifies the option to display Page Width button on the toolbar.
BTNP100PERCENT_VISIBLE	TRUE FALSE	Specifies the option to display 100% button on the toolbar.
ZOOMPERCENT_VISIBLE	TRUE FALSE	Specifies the option to display Zoom Level button on the toolbar.
BTNFIRST_VISIBLE	TRUE FALSE	Specifies the option to display First Page button on the toolbar.
BTNPRIOR_VISIBLE	TRUE FALSE	Specifies the option to display Previous Page button on the toolbar.
PAGENO_VISIBLE	TRUE FALSE	Specifies the option to display Page Number button on the toolbar.
BTNNEXT_VISIBLE	TRUE FALSE	Specifies the option to display Next Page button on the toolbar
BTNLAST_VISIBLE	TRUE FALSE	Specifies the option to display Last Page button on the toolbar.
BTNCANCEL_VISIBLE	TRUE FALSE	Specifies the option to display Cancel/Close button on the toolbar.
ZOOM_TYPE	PAGE_WIDTH WHOLE_PAGE 100_PERCENT PERCENTAGE	Specifies the zoom type. If PERCENTAGE is specified, the ZOOMPERCENT parameter must also be used so the percentage value is provided.
ZOOMPERCENT	value	Specifies the zoom level percentage, such as 70, 100, 120, 200, 250, etc. The ZOOM_TYPE value must be defined as PERCENTAGE in order to specify the ZOOMPERCENT value. See examples below.
PREVIEW_CAPTION	value	Specifies the caption for the preview window
PRINTER_NAME	value	Specifies the name of the default selection in the printers drop down of the print dialog, when the Printer button is visible

MDI		Specifies the option to display the preview window in a modeless window, to access other windows without closing it first.
LEFT	value	Specifies the left coordinate of the print preview window
TOP	value	Specifies the top coordinate of the print preview window
WIDTH	value	Specifies the width of the print preview window
HEIGHT	value	Specifies the height of the print preview window
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10- 12,20	Determines the pages in which the Page Style will appear on.
THEMENAME		Specifies one of 86 pre-defined Themes, or a custom Theme loaded into R:BASE. See Themes.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Examples:

-- Example 01 (Typical Print Preview on SCREEN with Toolbar):

```
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED+
|PREVIEW_CAPTION 'Typical Print Preview'
```

-- Example 02 (Typical Print Preview on SCREEN with zoom type as page width):

```
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|ZOOM_TYPE PAGE_WIDTH+
|PREVIEW_CAPTION 'Print Preview Using the Page Width Display'
```

-- Example 03 (Typical Print Preview on SCREEN with zoom type as whole page):

```
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|ZOOM_TYPE WHOLE_PAGE+
|PREVIEW_CAPTION 'Print Preview Using the Whole Page Display'
```

-- Example 04 (Typical Print Preview on SCREEN with zoom type as 100%):

```
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|ZOOM_TYPE 100_PERCENT+
|PREVIEW_CAPTION 'Print Preview Using the Percentage Display (100%)'
```

```
-- Example 05 (Typical Print Preview on SCREEN with zoom type as 75%):
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|ZOOM_TYPE PERCENTAGE|ZOOMPERCENT 75 +
|PREVIEW_CAPTION 'Print Preview Using the Percentage Display (75%)'

-- Example 06 (Print Preview Tool Bar without Without the Page Display Buttons):
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNWHOLEPAGE_VISIBLE FALSE+
|BTNPAGEWIDTH_VISIBLE FALSE+
|BTN100PERCENT_VISIBLE FALSE+
|ZOOMPERCENT_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Page Display Buttons'

-- Example 07 (Print Preview Tool Bar without the Navigation Buttons):
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNFIRST_VISIBLE FALSE+
|BTNPRIOR_VISIBLE FALSE+
|PAGENO_VISIBLE FALSE+
|BTNNEXT_VISIBLE FALSE+
|BTNLAST_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Navigation Buttons'

-- Example 08 (Print Preview Tool Bar without Printer Button):
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNPRINT_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Print Button'

-- Example 09 (Print Preview Tool Bar without Cancel/Close Button)
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|BTNCANCEL_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Cancel/Close Button'

-- Example 10 (Print Preview without the entire Tool Bar)
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|WINDOW_STATE MAXIMIZED +
|TOOLBAR_VISIBLE FALSE+
|PREVIEW_CAPTION 'Print Preview Without the Entire Toolbar'

-- Example 11 (Print Preview with a MDI window using specified coordinates)
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN MDI|LEFT 20|TOP 20|WIDTH 800|HEIGHT 500

-- Example 12 (Print Preview with a different printer specified for the Print Dialog)
PRINT Invoice WHERE TransID = 1002 +
OPTION SCREEN|PRINTER_NAME 'HP LaserJet 1018'
```

1.17.6.2 PRINTER

The PRINTER output option sends the report to a printer.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION PRINTER
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TRAY		Specifies the name of the bin (paper tray) containing the paper on which the report will be printed. A common use of this property is to set it to Manual Feed on mailing label reports. That way, you can provide a means to load the special label paper without manually setting the printer.
COLLATION	ON OFF	Indicates whether multiple copies will print in pre-sorted order
COPIES	value	Specifies the number of copies to print. If the COLLATION property is set to ON, the copies will print in sorted order.
DOCUMENT_NAME	value	When a report is printing, this is the name of the document as it should appear when listed in the Print Manager and on network header pages. This value can also be displayed in reports using the DocumentName type of the System Variable control.
DUPLEX	NONE HORIZONTAL VERTICAL	Indicates what type of two-sided printing should occur. Not all printers support duplex printing. Horizontal duplex printing is performed across the page, from left to right, so the text will read as in a book. Vertical duplex printing is performed down the page, from top to bottom, so the text will read as a tablet.
ORIENTATION	PORTRAIT LANDSCAPE	Determines whether a report prints vertically (PORTRAIT) or horizontally (LANDSCAPE) on a page.
PAPER_NAME	value	The name of the paper on which the report will be printed. When you specify a value for the PAPER_NAME, the width and height are automatically set to the proper values. For example, if "Legal" is specified, the width will be set to 8.5" and the height will be set to 14".
PAPER_WIDTH		The value of the PAPER_WIDTH is always expressed in the units specified by the report.
PAPER_HEIGHT		The value of the PAPER_HEIGHT is always expressed in the units specified by the report.
PRINTER_NAME		The name of the printer that will be used to print the report.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING	ALL FIRST LAST Page List e.g. 1,4-6,10- 12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is

		set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Examples:

--Example 01 (To print Customer List on your default Windows printer):

```
PRINT CustomerList OPTION PRINTER
```

--Example 02 (To print Customer List on your network printer):

```
PRINT CustomerList +
OPTION PRINTER +
|COPIES 1 +
|ORIENTATION PORTRAIT +
|PRINTER_NAME \\MyServer\hp LaserJet 1320 PCL 6
```

--Example 03 (To print Customer Shipping Labels on your label printer):

```
LBLPRINT CustShippingLabel WHERE CustID = 127 OPTION PRINTER +
|COPIES 2 +
|PRINTER_NAME DYMO LabelWriter 2 +
|ORIENTATION LANDSCAPE
```

--Example 04 (To print first page of Customer List report only)

```
PRINT CustomerList OPTION PRINTER|PAGE_SETTINGS FIRST
```

--Example 05 (To print last page of Customer List Report only)

```
PRINT CustomerList OPTION PRINTER|PAGE_SETTINGS LAST
```

--Example 06 (To PRINT all pages)

```
PRINT ReportName OPTION PRINTER|PAGE_SETTINGS ALL
```

--Example 07 (To print pages 3, 5, and 8 of your big report)

```
PRINT ReportName OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,5,8
```

--Example 08 (To print pages 3 and 8 of your big report)

```
PRINT ReportName OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,8
```

--Example 09 (To print page 3 only)

```
PRINT ReportName OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3
```

--Example 10 (To print pages 1 through 3 and 8 of your big report)

```
PRINT ReportName OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 1-3,8
```

--Example 11 (To print pages 3, 5, and 9 through 15 of your big report)

```
PRINT ReportName OPTION PRINTER +
|PAGE_SETTINGS PAGE_LIST +
|PAGE_LIST_VALUES 3,5,9-15
```

1.17.6.3 BMP

The BMP output option displays the report in a Bitmap image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION BMP
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether the file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.

MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as BMP file
PRINT Invoice WHERE TransID = 1068 +
OPTION BMP +
|FILENAME Invoice1068.BMP +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.17.6.4 DIF

The DIF output option displays the report in a Data Interchange Format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION DIF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.

AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as DIF file
PRINT Invoice WHERE TransID = 1068 +
OPTION DIF +
|FILENAME Invoice1068.DIF +
|SHOW_CANCEL_DIALOG OFF +
|OPEN OFF +
|ROW_SPACING 1
```

1.17.6.5 EMF

The EMF output option displays the report in an Enhanced Metafile image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION EMF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).

MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as EMF file
PRINT Invoice WHERE TransID = 1068 +
OPTION EMF +
|FILENAME Invoice1068.EMF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.17.6.6 ETXT

The ETXT output option displays the report in a report emulation text file, which prints the report as plain readable text including all bands, such as Report Header, Page Header, Break Headers, Break Footers, Page Footers and Report Footers, if exist.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION ETXT
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
APPEND		Appends the report data to the specified FILENAME value.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to

		the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Examples:

--To print columnar report as Emulation Text - ETXT file

```
PRINT ColumnarReport +
OPTION ETXT +
|FILENAME ColumnarReport.TXT +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON
```

--Prints the BonusText report and appends the output to the BonusText.txt file

```
PRINT BonusText +
OPTION ETXT +
|FILENAME BonusText.txt +
|APPEND
```

1.17.6.7 GIF

The GIF output option displays the report in a Graphics Interchange Format image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION GIF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT	Specifies the way background image has to be displayed in the GIF file.

	TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as GIF file
PRINT Invoice WHERE TransID = 1068 +
OPTION GIF +
|FILENAME Invoice1068.GIF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.17.6.8 HTML

The HTML output option displays the report in a hypertext markup language file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION HTML
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the way background image has to be displayed in the file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TY PE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.

INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.

EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON
-------	-----------	---

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed HTML parameter options:

OPTIMIZE_FOR_IE	ON OFF	Specifies whether the entire report output as HTML document should be optimized for the MS Internet Explorer browser.
PAGE_END_LINES	ON OFF	Indicates whether a line should be drawn at the end of each page. Setting PAGE_END_LINES to ON draws a line after each page. PAGE_END_LINES is considered only when SEP_FILE_PER_PAGE is set to OFF.
OUTPUT_STYLES_TO_CSS_FILE	ON OFF	Specifies whether to output styles to a CSS (Cascade Style Sheet) file.
CSSFILE	value	Specifies the CSS file name if OUTPUT_STYLES_TO_CSS_FILE is set to ON.
OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE	ON OFF	Specifies whether to output scripts to a Java Script File.
JAVA_SCRIPT_FILE_NAME	value	Specifies the file name if the OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE is set to ON.
GENERATE_TOC	ON OFF	Specifies whether to generate a Table of Contents based on all report Break Header(s), if defined.
TOC_FILE_NAME	value	Specifies the file name for the Table of Contents, if the GENERATE_TOC option is set to ON.
IMAGE_FOLDER	value	specify the folder path and name for HTML document images.
DEFAULT_FONT_NAME	value	Specifies the default font name for HTML document.
DEFAULT_FONT_SIZE	value	Specifies the default font size for HTML document.
DEFAULT_FONT_COLOR	value	Specifies the default font color for HTML document.
DEFAULT_FONT_BOLD	ON OFF	Specifies whether the font is bold faced
DEFAULT_FONT_ITALIC	ON OFF	Specifies whether the font is italicized.
DEFAULT_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined.
DEFAULT_FONT_STRIKEOUT	ON OFF	Specifies whether the font is displayed with a horizontal line through it.
SEP_FILE_PER_PAGE	ON OFF	Specifies whether to create a separate file for each page of the report.
NAV_SHOW	ON OFF	Set to ON to show a collection of links that enable easy navigation between pages of the exported HTML. The links function much like the navigator buttons in the standard preview of Report Preview: one link to jump to the first page; one to the last page; one to the next page and another to get back to the previous page. NAV_SHOW is only effective when the option for SEP_FILE_PER_PAGE is set to ON.

LINK_BACK_COLOR	value	Specifies the background color of navigator. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_BACK_COLOR	value	Specifies the background color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_FORE_COLOR	value	Specifies the color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
USE_TEXT_LINKS	ON OFF	Specifies the type of the navigator links that should be encoded for a multi-file HTML output. Setting USE_TEXT_LINKS to ON uses the text provided in LINK_TEXT_FIRST, LINK_TEXT_PREV, LINK_TEXT_NEXT and LINK_TEXT_LAST as the hyperlink's caption. Setting to OFF uses Images specified by LINK_IMG_FILE_FIRST, LINK_IMG_FILE_PREV, LINK_IMG_FILE_NEXT, LINK_IMG_FILE_LAST. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
LINK_TEXT_FIRST	value	Specifies the hyperlink's text caption for first page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_PREV	value	Specifies the hyperlink's text caption for previous page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_NEXT	value	Specifies the hyperlink's text caption for next page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_LAST	value	Specifies the hyperlink's text caption for last page when USE_TEXT_LINKS option is set to ON.
LINK_FONT_NAME	value	Specifies the font name for navigator links.
LINK_FONT_SIZE	value	Specifies the font size for navigator links.
LINK_FONT_COLOR	value	Specifies the font color for navigator links.
LINK_FONT_BOLD	ON OFF	Specifies whether the font is bold faced for navigator links.
LINK_FONT_ITALIC	ON OFF	Specifies whether the font is italicized for navigator links.
LINK_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined for navigator links.
LINK_FONT_STRIKEOUT	ON OFF	Specifies whether the font for navigator link is displayed with a horizontal line through it.
LINK_IMG_FILE_FIRST	value	Specifies the hyperlink's image file for first page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_PREV	value	Specifies the hyperlink's image file for previous page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_NEXT	value	Specifies the hyperlink's image file for next page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_LAST	value	Specifies the hyperlink's image file for last page when USE_TEXT_LINKS option is

		set to OFF.
NAV_TYPE	PAGE SCREEN	Specifies the type of the navigator. If NAV_TYPE is set to SCREEN then the navigator will be fixed to screen, else the navigator will be fixed to page. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_ORIENT	VERTICAL HORIZONTAL	Specifies the orientation of the navigator. if NAV_ORIENT is set to HORIZONTAL then, a Horizontal navigator is encoded, else a Vertical navigator is encoded. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_POS	TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the position of the navigator. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
DISABLE_RIGHT_CLICK	ON OFF	Specifies whether to disable the typical speed menu options when you right click on your mouse button.
DISABLE_COPYING	ON OFF	Specifies whether to disable the option to highlight and copy the text.
AUTO_MAXIMIZE_WINDOW	ON OFF	Specifies whether to automatically maximize the window when HTML document is opened.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print SalesSumAndSubTot report as HTML file
PRINT SalesSumAndSubTot +
OPTION HTML +
|FILENAME SalesSumAndSubTot.htm +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
|INCLUDE_RICHTEXT OFF +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|IMAGE_DPI -1 +
|PIXELFORMAT 24 +
|INCLUDE_HYPERLINKS ON +
|TITLE Customer Sales Sub-Totals and Totals +
|SUBJECT ConComp Sales Summary Report +
|AUTHOR Accounting Team of ConComp +
|KEYWORDS SubTotals Totals Sales Summary +
```

```

|OPTIMIZE_FOR_IE ON +
|PAGE_END_LINES ON +
|OUTPUT_STYLES_TO_CSS_FILE ON +
|CSSFILE +
|OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE ON +
|JAVA_SCRIPT_FILE_NAME +
|GENERATE_TOC ON +
|TOC_FILE_NAME Index.htm +
|IMAGE_FOLDER Images +
|DEFAULT_FONT_NAME Verdana +
|DEFAULT_FONT_SIZE 9 +
|DEFAULT_FONT_COLOR BLACK +
|DEFAULT_FONT_BOLD OFF +
|DEFAULT_FONT_ITALIC OFF +
|DEFAULT_FONT_UNDERLINE OFF +
|DEFAULT_FONT_STRIKEOUT OFF +
|SEP_FILE_PER_PAGE ON +
|NAV_SHOW ON +
|LINK_BACK_COLOR WHITE +
|LINK_H_BACK_COLOR BLUE +
|LINK_H_FORE_COLOR WHITE +
|USE_TEXT_LINKS OFF +
|LINK_TEXT_FIRST First +
|LINK_TEXT_PREV Previous +
|LINK_TEXT_NEXT Next +
|LINK_TEXT_LAST Last +
|LINK_FONT_NAME Wingdings +
|LINK_FONT_SIZE 10 +
|LINK_FONT_COLOR BLUE +
|LINK_FONT_BOLD OFF +
|LINK_FONT_ITALIC OFF +
|LINK_FONT_UNDERLINE OFF +
|LINK_FONT_STRIKEOUT OFF +
|LINK_IMG_FILE_FIRST Images\FirstOff.gif +
|LINK_IMG_FILE_PREV Images\PrevOff.gif +
|LINK_IMG_FILE_NEXT Images\NextOff.gif +
|LINK_IMG_FILE_LAST Images\LastOff.gif +
|NAV_TYPE PAGE +
|NAV_ORIENT HORIZONTAL +
|NAV_POS BOTTOMCENTER +
|DISABLE_RIGHT_CLICK OFF +
|DISABLE_COPYING OFF +
|AUTO_MAXIMIZE_WINDOW ON +
|OPEN ON

```

1.17.6.9 JPG

The JPG output option displays the report in a JPEG image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION JPG
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
-----------	-------	-------------

FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
GRAYSCALE	ON OFF	Determines whether the image output of a JPEG image is black and white, or color. Use GRAYSCALE for reading in and writing out, that is, for decompressing and compressing, a JPEG image. This property affects how the image is displayed. GRAYSCALE is used for speed in output optimization. When GRAYSCALE is set to ON, the color is separated from luminosity. Color takes the most time to decompress, so for previewing the image, GRAYSCALE can be set to True for speed. The output then contains 255 shades of gray.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
PROGRESSIVE_ENCODING	ON OFF	Determines whether an image can be progressively displayed when it is decompressed.
QUALITY	1-100	Determines the closeness of the saved, JPEG image to the original. As a side effect it also matter to the size of the saved file. Setting a larger value improves image quality but also increases file size; a smaller value decreases image quality and also reduces file size. Alter it according to the need.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST	Determines the pages in which the Page Style will appear on.

	LAST Page List e.g. 1,4-6,10-12,20	
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as JPG file
PRINT Invoice WHERE TransID = 1068 +
OPTION JPG +
|FILENAME Invoice1068.JPG +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|GRAYSCALE OFF +
|PIXELFORMAT 32 +
|QUALITY 100 +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.17.6.10 LOTUS

The LOTUS output option displays the report in a Lotus 1-2-3 format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION LOTUS
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.

SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as Lotus file
PRINT Invoice WHERE TransID = 1068 +
OPTION LOTUS +
|FILENAME Invoice1068.WK1 +
|SHOW_CANCEL_DIALOG ON +
```

```
| OPEN OFF +
| ROW_SPACING 1
```

1.17.6.11 PDF

The PDF output option displays the report in a Portable Document Format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION PDF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTE R BOTTOMRIGHT	Specifies the way background image has to be displayed in the PDF file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TY PE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.
INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value (0-100)	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.

PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed PDF parameter options:

GENERATE_TOC	ON OFF	Specifies whether to generate bookmarks or a "Table of Contents" based on all report Break Headers, if defined.
USE_COMPRESSION	ON OFF	Specifies whether compression should be used.

COMPRESSION_METHOD	FASTEST NORMAL MAXCOMPRESS	Specifies the type of compression to be used to compress text and Image.
FONT_ENCODING	WIN_ANSI MAC_ROMAN MAC_EXPERT STANDART PDF_DOC	Specifies the type of font encoding to be used to encode font.
EMBED_USED_FONTS	ON OFF FULL (same as "ON" for backward compatibility) SUBSET	Specifies whether all the used fonts should be embedded in the PDF document. Set EMBED_USED_FONTS to ON to embed all the fonts used in the report. Fonts are embedded only if embedding is permitted depending on the Embedding Flags of the font.
ENCRYPT_FILE	ON OFF (default)	Specifies whether the PDF document should be encrypted.
USER_PASSWORD	value	Specifies the user password of the encrypted PDF document. Opening the document with the correct user password allows additional operations to be performed according to the user access permissions specified by the permissions properties such as, ENABLE_PRINTING (ON/OFF), ENABLE_COPYING (ON/OFF), and ENABLE_MODIFYING (ON/OFF). These permissions are considered by the viewer applications if the document is opened with the User password. If ENABLE_PRINTING is OFF then printing is not enabled. If ENABLE_COPYING is OFF then page contents cannot be selected and copied. If ENABLE_MODIFYING is OFF then the PDF document cannot be modified.
OWNER_PASSWORD	value	Specifies the owner password of the encrypted PDF document. Opening the document with the correct owner password (assuming it is not same as the user password) allows full (owner) access to the document. This unlimited access includes the ability to change the document's passwords and access permissions.
ENCRYPTION_LEVEL	40BIT 128BIT	Specifies the level of encryption of the PDF document. Set ENCRYPTION_LEVEL to 40BIT for a 40 bit encryption of the PDF document, i.e a 40 bit key is used to encrypt the PDF document. Set ENCRYPTION_LEVEL to 128BIT for a 128 bit encryption of the PDF document, i.e a 128 bit key is used to encrypt the PDF

		document. ENCRYPTION_LEVEL is considered only if ENCRYPT_FILE is set to ON.
ENABLE_PRINTING	ON (default) OFF	Specifies whether to allow printing when USER_PASSWORD is set.
ENABLE_COPYING	ON (default) OFF	Specifies whether to allow page contents to be selected and copied when USER_PASSWORD is set.
ENABLE_MODIFYING	ON (default) OFF	Specifies whether the PDF document can be modified when USER_PASSWORD is set.
ENABLE_ANNOTATION	ON OFF	Specifies whether the PDF document supports annotation when USER_PASSWORD is set.
ENABLE_FORM_FILL	ON OFF	Specifies whether the PDF document supports form fill when USER_PASSWORD is set.
ENABLE_ACCESSIBILITY	ON OFF	Specifies whether the PDF document supports tagging accessibility when USER_PASSWORD is set.
ENABLE_DOC_ASSEMBLY	ON OFF	Specifies whether the PDF document supports document assembly when USER_PASSWORD is set.
ENABLE_HIRES_PRINT	ON OFF	Specifies whether the PDF document supports high resolution printing when USER_PASSWORD is set.
HIDE_TOOLBAR	ON OFF (default)	Specifies whether to hide the viewer application's toolbars, such as Acrobat Reader, when the document is active. If the value is ON then the viewer applications toolbars are hidden.
HIDE_MENUBAR	ON OFF (default)	Specifies whether to hide the viewer application's menu bar, such as Acrobat Reader, when the document is active. If the value is ON then the viewer applications menu bar is hidden.
HIDE_WINDOW_UI	ON OFF (default)	Specifies whether to hide user interface elements in the document's window (such as scroll bars and navigation controls), leaving only the document's contents displayed. If the value is ON then documents user interface elements are hidden.
FIT_WINDOW	ON OFF (default)	Specifies whether to resize the document's window to fit the size of the first displayed page. If the value is ON then the first page displayed is resized to fit the document's window.
CENTER_WINDOW	ON OFF (default)	Specifies whether to position the document's window in the center of the screen. If the value is ON then the document is positioned at the center of the screen when opened.

PAGE_LAYOUT	SINGLE_PAGE ONE_COLUMN TWO_COLUMN_LEFT TWO_COLUMN_RIGHT	Specifies the page layout to be used when the PDF document is opened. If the value is set to SINGLE_PAGE then one page is displayed at a time. If the value is set to ONE_COLUMN then pages are displayed in one column. If the value is set to TWO_COLUMN_LEFT then pages are displayed in two columns with odd numbered pages on the left. If the value is set to TWO_COLUMN_RIGHT then pages are displayed in two columns with odd numbered pages on the right.
PAGE_MODE	NONE OUTLINES THUMBS FULLSCREEN	Specifies how the document should be displayed when opened. If the value is set to NONE then neither document outlines nor thumbnail images are visible. If the value is set to OUTLINES then document outline is visible. If the value is set to THUMBS then thumbnail images are visible. If the value is set to FULLSCREEN then document is opened in Full Screen mode with no menu bar, window controls, or any other window visible.
NON_FULL_SCREEN_PAGE_MODE	NONE USE_OUTLINES USE_THUMB	Specifies how to display the document on exiting full screen mode. This is meaningful only if the PAGE_MODE value is set to FULLSCREEN. If the value is set to NONE then neither document outlines nor thumbnail images are visible. If the value is set to USE_OUTLINES then document outline is visible. If the value is set to USE_TNUMB then thumbnail images are visible.
PRESENTATION_MODE	SPLIT_HOR_INWARD SPLIT_HOR_OUTWARD SPLIT_VERT_INWARD SPLIT_VERT_OUTWARD BLINDS_HOR BLINDS_VERT BOX_INWARD BOX_OUTWARD WIPE_LEFT_TO_RIGHT WIPE_BOTTOM_TO_TOP WIPE_RIGHT_TO_LEFT WIPE_TOP_TO_BOTTOM DISSOLVE GLITTER_LEFT_TO_RIGHT GLITTER_TOP_TO_BOTTOM GLITTER_TOPLEFT_TO_BOTTOMRIGHT NONE	Specifies how the PDF viewer application will display/present the document with transition effects. Some viewer applications allow a document to be displayed in the form of a presentation or "slide show," advancing from one page to the next either automatically or under user control. A detailed description of each transition effect is described below.
DURATION	value	Specifies the duration of the presentation mode (transition effect) in seconds.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Transition Effects

SPLIT_HOR_INWARD	Two lines sweep across the screen, revealing the new page. The lines are horizontal and move inward from the edges of the page.
SPLIT_HOR_OUTWARD	Two lines sweep across the screen, revealing the new page. The lines are horizontal and move outward from the center of the page.
SPLIT_VERT_INWARD	Two lines sweep across the screen, revealing the new page. The lines are vertical and move inward from the edges of the page.
SPLIT_VERT_OUTWARD	Two lines sweep across the screen, revealing the new page. The lines are vertical and move outward from the center of the page.
BLINDS_HOR	Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are horizontal and move downward.
BLINDS_VERT	Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines are vertical and move to the right.
BOX_INWARD	A rectangular box sweeps inward from the edges of the page revealing the new page.
BOX_OUTWARD	A rectangular box sweeps outward from the center of the page revealing the new page.
WIPE_LEFT_TO_RIGHT	A single line sweeps across the screen from Left to Right edge of the page.
WIPE_BOTTOM_TO_TOP	A single line sweeps across the screen from Bottom to the Top edge of the page.
WIPE_RIGHT_TO_LEFT	A single line sweeps across the screen from Right to the Left edge of the page.
WIPE_TOP_TO_BOTTOM	A single line sweeps across the screen from Top to Bottom edge of the page.
DISSOLVE	The old page "dissolves" gradually to reveal the new one.
GLITTER_LEFT_TO_RIGHT	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Left side of the screen to the Right side.
GLITTER_TOP_TO_BOTTOM	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Top side of the screen to the Bottom side.
GLITTER_TOPLEFT_TO_BOTTOMRIGHT	Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from Top Left corner of the screen to the Bottom Right corner.
NONE	The new page simply replaces the old one with no special transition effect.

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print SalesSumAndSubTot report as secure PDF document
PRINT SalesSumAndSubTot +
OPTION PDF +
|FILENAME SalesSumAndSubTotals.PDF +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
```

```

|INCLUDE_RICHTEXT ON +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
|IMAGE_FORMAT JPG +
|PIXELFORMAT 32 +
|JPEG_QUALITY 100 +
|IMAGE_DPI -1 +
|INCLUDE_HYPERLINKS ON +
|GENERATE_TOC ON +
|TITLE Customer Sales Sub-Totals and Totals +
|SUBJECT Sales Summary Report +
|AUTHOR Accounting Team of Consolidated Computer Company +
|KEYWORDS SubTotals Totals Sales Summary +
|USE_COMPRESSION ON +
|COMPRESSION_METHOD MAXCOMPRESS +
|FONT_ENCODING WIN_ANSI +
|EMBED_USED_FONTS ON +
|ENCRYPT_FILE ON +
|USER_PASSWORD ATCCC +
|OWNER_PASSWORD ATCCOwner +
|ENCRYPTION_LEVEL 128BIT +
|ENABLE_PRINTING OFF +
|ENABLE_COPYING OFF +
|ENABLE_MODIFYING OFF +
|HIDE_TOOLBAR OFF +
|HIDE_MENUBAR OFF +
|HIDE_WINDOW_UI OFF +
|FIT_WINDOW OFF +
|CENTER_WINDOW OFF +
|PAGE_LAYOUT SINGLE_PAGE +
|PAGE_MODE NONE +
|NON_FULL_SCREEN_PAGE_MODE NONE +
|PRESENTATION_MODE NONE +
|DURATION 0 +
|OPEN ON

```

1.17.6.12 PNG

The PNG output option displays the report in a Portable Network Graphics image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION PNG
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report
INCLUDE_TEXT	ON OFF	Specifies whether text on the report should be included in the exported file.

INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is - 1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
COLOR_TYPE	GRAYSCALE RGB (default) RGB_ALPHA GRAY_ALPHA	Specifies the color type of the image
PIXEL_DIMENSIONS	ON OFF	Enables the ability to set PNG pixel dimension information
UNIT_SPECIFIER	NONE METER	Specifies the units for setting the pixel dimension information. PIXEL_DIMENSIONS must be set ON in order to use the parameter. With NONE, X_PIXELS_PER_UNIT and Y_PIXELS_PER_UNIT defines pixel aspect ratio only, and the actual size of the pixels remains unspecified. Using METER the actual size of the pixels is specified
X_PIXELS_PER_UNIT	value	Specifies the number of pixels per unit in the horizontal direction. PIXEL_DIMENSIONS must be set ON in order to use the parameter. Default: 1
Y_PIXELS_PER_UNIT	value	Specifies the number of pixels per unit in the vertical direction. PIXEL_DIMENSIONS must be set ON in order to use the parameter. Default: 1
TRANSPARENCY	ON OFF	Sets the fields related to transparency of the output
IMAGE_ALPHA	0 - 100	Specifies a percentage of transparency for the output. This is applicable only when TRANSPARENCY is ON and COLOR_TYPE is either RGB_ALPHA or GRAY_ALPHA. If IMAGE_ALPHA is 0, then the image is completely transparent. If IMAGE_ALPHA is 100, then the image is completely opaque. Default: 0
TRANSPARENT_COLOR	value	Specifies the color that is transparent. This is applicable only when TRANSPARENCY is ON and COLOR_TYPE is either RGB or GRAYSCALE.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then

		that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as PNG file
PRINT Invoice WHERE TransID = 1068 +
OPTION PNG +
|FILENAME Invoice1068.PNG +
|SHOW_CANCEL_DIALOG OFF +
|INCLUDE_SHAPES OFF +
|INCLUDE_IMAGES ON +
|OPEN ON
```

1.17.6.13 QPRO

The QPRO output option displays the report in a Quattro Pro format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION QPRO
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.

SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as Quattro Pro file
PRINT Invoice WHERE TransID = 1068 +
OPTION QPRO +
|FILENAME Invoice1068.WB1 +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON +
|COLUMN_SPACING 1
```

1.17.6.14 RTF

The RTF output option displays the report in a rich text format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION RTF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
APPEND		Appends the report data to the specified FILENAME value.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
GRAPHIC_DATA_IN_BINARY	ON (default) OFF	Specifies whether graphic data is to be encoded in binary format. Set to ON to encode graphic data in binary format. If OFF, graphic data is encoded in hexadecimal format.
OPTIMIZATION	POSITIONAL_ACCURACY (default) EASE_OF_EDITING	Specifies the type of encoding of the RTF document. Resulting RTF document can be either optimized for positional accuracy (meaning that the placement of the objects will be as close as possible to a printed report), or for ease of editing (meaning that everything is treated as simple text and it is much easier to edit the file at the expense of aesthetics). Set

		OPTIMIZATION EASE_OF_EDITING if you wish to easily modify the resulting RTF document.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content in RTF document property summary.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print invoice report as RTF document
PRINT Invoice WHERE TransID = 1068 +
OPTION RTF +
```

```

FILENAME Invoice1068.RTF +
SHOW_CANCEL_DIALOG ON +
BACKGROUND_FILE NONE +
BACKGROUND_TYPE CENTER +
INCLUDE_LINES ON +
INCLUDE_SHAPES ON +
INCLUDE_RICHTEXT ON +
INCLUDE_IMAGES ON +
IMAGE_FORMAT JPG +
JPEG_QUALITY 100 +
IMAGE_DPI -1 +
PIXELFORMAT 32 +
GRAPHIC_DATA_IN_BINARY ON +
OPTIMIZATION POSITIONAL_ACCURACY +
TITLE Customer Invoice +
SUBJECT Sales Order +
AUTHOR Accounting Team of Consolidated Computer Company +
KEYWORDS Invoice +
OPEN ON

```

1.17.6.15 SVG

The SVG output option displays the report in a scalable vector graphics file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SVG
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report
INCLUDE_TEXT	ON OFF	Specifies whether text on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
IMAGE_FORMAT	GIF JPG BMP	Specifies the image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.

JPEG_QUALITY	1-100	Specifies the quality of the JPEG Image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as SVG file
PRINT Invoice WHERE TransID = 1068 +
OPTION SVG +
|FILENAME Invoice1068.SVG +
|SHOW_CANCEL_DIALOG OFF +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|OPEN ON
```

1.17.6.16 SYLK

The SYLK output option displays the report in a Symbolik Link format file.

Syntax:

PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION SYLK

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
COLUMN_SPACING	value	Specifies the column spacing between data. Use -1 for auto-spacing.
ROW_SPACING	value	Specifies the row spacing between data. Use -1 for auto-spacing.
AUTO_FORMAT_CELLS	ON OFF	Specifies if the format of the rendered text (Integer, Double etc) is detected automatically. Otherwise the content is just inserted as text without any format considerations.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as Symbolik Link format file
PRINT Invoice WHERE TransID = 1070 +
OPTION SYLK +
|FILENAME Invoice1070.sylk +
|SHOW_CANCEL_DIALOG ON +
|OPEN ON +
|ROW_SPACING 1
```

1.17.6.17 TIFF

The TIFF output option displays the report in the tagged image file format.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION TIFF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
MONOCHROME	ON OFF	Specifies whether the file produced are color or monochrome images.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the file (bits/pixel).
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
MULTIPAGE	ON OFF	Specifies the output report to one multi-page file or one file per page.
COMPRESSION	NONE PACKBITS GROUP3_1D GROUP3_2D GROUP4 ZLIB JPEG	When an image is loaded, this parameter indicates the compression used by the image. When saving in image, the parameter may be set to specify which compression method to use.

OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as TIF file
PRINT Invoice WHERE TransID = 1068 +
OPTION TIFF +
|FILENAME Invoice1068.TIF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|MONOCHROME OFF +
|MUTIPAGE OFF +
|PIXELFORMAT 32 +
|SCALEX 1 +
|SCALE 1 +
|COMPRESSION NONE +
|OPEN ON
```

1.17.6.18 TXT

The TXT output option displays the report in a text file format.

Caution:

This text file option requires the report bands to be defined within the Report Designer. From the main Menu Bar, select "File" > "Print to Text File Setup..."

If a user attempts to print the report as a text file using the OPTION TXT parameter without first having defined the "Text File Setup" within the Report Designer, an error message will be displayed and a zero length file will be created. The same rule applies when printing a report to a text file from the Database Explorer.

For step-by-step instructions on how to set up a report to print as Text File, please see Printing Report as Text File.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION TXT
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.

EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON
-------	-----------	--

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.
- The TXT "Text File" option allows you to define a particular band to print the data as Comma Delimited, Tab Delimited or Fixed Length file. This is the perfect option to export selective and/or summarized data used in a report band.

Example:

```
--To print columnar report as a Text file
PRINT ColumnarReport +
OPTION TXT +
|FILENAME ColumnarReport.TXT +
|SHOW_CANCEL_DIALOG OFF +
|OPEN ON
```

1.17.6.19 WMF

The WMF output option displays the report in a Windows Metafile image file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION WMF
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the BMP file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
SCALEX	value	Scaling factor for the width of the exported page. Use 1 for actual width.
SCALEY	value	Scaling factor for the height of the exported page. Use 1 for actual height.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for

	PAGE_LIST	PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTING S	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print Invoice as WMF file
PRINT Invoice WHERE TransID = 1068 +
OPTION WMF +
|FILENAME Invoice1068.WMF +
|SHOW_CANCEL_DIALOG OFF +
|BACKGROUND_COLOR WHITE +
|SCALEX 1 +
|SCALE 1 +
|OPEN ON
```

1.17.6.20 XHTML

The XHTML output option displays the report in a extensible hypertext markup language file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XHTML
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
BACKGROUND_COLOR	value	Indicates the color to be used as background color of all pages of report.
BACKGROUND_TYPE	TILE TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the way background image has to be displayed in the file.
BACKGROUND_FILE	value	Stores the image file name that is to be used as background image of all pages of report.
INCLUDE_LINES	ON OFF	Specifies whether lines on the report should be included in the exported file.
INCLUDE_SHAPES	ON OFF	Specifies whether shapes on the report should be included in the exported file.
INCLUDE_RICHTEXT	ON OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TYPE	IMAGE PLAINTEXT	Specifies the way in which Rich Text contents should be encoded in exported HTML document when INCLUDE_RICHTEXT option is set to ON. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode Rich Text contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode Rich Text contents as unformatted text. The font used when encoding Rich Text as plain text is the one set in the Font property of the Rich Text object.
INCLUDE_HYPERLINKS	ON OFF	Specifies whether hyperlinks in your report should be included in the exported file.
INCLUDE_IMAGES	ON OFF	Specifies whether images on the report should be included in the exported file.
IMAGE_FORMAT	GIF JPG BMP	Specifies the linked image file format for image controls used in the report when INCLUDE_IMAGES option is set to ON.
JPEG_QUALITY	value 0 - 100	Specifies the quality of the JPEG image when INCLUDE_IMAGES is set to ON and IMAGE_FORMAT is set to JPG.
IMAGE_DPI	value	Specifies the depth of image as "dots per inch" to enhance the image. By increasing this value, the file size will increase. An example would be 200. The Default Value is -1.
PIXELFORMAT	1 : 1bit 4 : 4bit 8 : 8bit 15 : 15bit 16 : 16bit 24 : 24bit 32 : 32bit	Specifies the pixel format of the image file (bits/pixel).
TITLE	value	Specifies the title of the report
SUBJECT	value	Specifies the subject of the report
AUTHOR	value	Specifies the author of the report
KEYWORDS	value	Stores the list of keywords used in the report. Specify a list of keywords for the document that should be encoded as the Keywords Meta content.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.

PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Detailed HTML parameter options:

PAGE_END_LINES	ON OFF	Indicates whether a line should be drawn at the end of each page. Setting PAGE_END_LINES to ON draws a line after each page. PAGE_END_LINES is considered only when SEP_FILE_PER_PAGE is set to OFF.
OUTPUT_STYLES_TO_CSS_FILE	ON OFF	Specifies whether to output styles to a CSS (Cascade Style Sheet) file.
CSSFILE	value	Specifies the CSS file name if OUTPUT_STYLES_TO_CSS_FILE is set to ON.
OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE	ON OFF	Specifies whether to output scripts to a Java Script File.
JAVA_SCRIPT_FILE_NAME	value	Specifies the file name if the OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE is set to ON.
GENERATE_TOC	ON OFF	Specifies whether to generate a Table of Contents based on all report Break Header(s), if defined.

TOC_FILE_NAME	value	Specifies the file name for the Table of Contents, if the GENERATE_TOC option is set to ON.
IMAGE_FOLDER	value	specify the folder path and name for XHTML document images.
DEFAULT_FONT_NAME	value	Specifies the default font name for XHTML document.
DEFAULT_FONT_SIZE	value	Specifies the default font size for XHTML document.
DEFAULT_FONT_COLOR	value	Specifies the default font color for XHTML document.
DEFAULT_FONT_BOLD	ON OFF	Specifies whether the font is bold faced
DEFAULT_FONT_ITALIC	ON OFF	Specifies whether the font is italicized.
DEFAULT_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined.
DEFAULT_FONT_STRIKEOUT	ON OFF	Specifies whether the font is displayed with a horizontal line through it.
SEP_FILE_PER_PAGE	ON OFF	Specifies whether to create a separate file for each page of the report.
NAV_SHOW	ON OFF	Set to ON to show a collection of links that enable easy navigation between pages of the exported XHTML. The links function much like the navigator buttons in the standard preview of Report Preview: one link to jump to the first page; one to the last page; one to the next page and another to get back to the previous page. NAV_SHOW is only effective when the option for SEP_FILE_PER_PAGE is set to ON.
LINK_BACK_COLOR	value	Specifies the background color of navigator. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_BACK_COLOR	value	Specifies the background color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
LINK_H_FORE_COLOR	value	Specifies the color of navigator when the mouse is hovered over it. This option will only be used if the SEP_FILE_PER_PAGE and NAV_SHOW options are set to ON.
USE_TEXT_LINKS	ON OFF	Specifies the type of the navigator links that should be encoded for a multi-file HTML output. Setting USE_TEXT_LINKS to ON uses the text provided in LINK_TEXT_FIRST, LINK_TEXT_PREV, LINK_TEXT_NEXT and LINK_TEXT_LAST as the hyperlink's caption. Setting to OFF uses Images specified by LINK_IMG_FILE_FIRST, LINK_IMG_FILE_PREV, LINK_IMG_FILE_NEXT, LINK_IMG_FILE_LAST. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
LINK_TEXT_FIRST	value	Specifies the hyperlink's text caption for first page when USE_TEXT_LINKS option is set to ON.

LINK_TEXT_PREV	value	Specifies the hyperlink's text caption for previous page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_NEXT	value	Specifies the hyperlink's text caption for next page when USE_TEXT_LINKS option is set to ON.
LINK_TEXT_LAST	value	Specifies the hyperlink's text caption for last page when USE_TEXT_LINKS option is set to ON.
LINK_FONT_NAME	value	Specifies the font name for navigator links.
LINK_FONT_SIZE	value	Specifies the font size for navigator links.
LINK_FONT_COLOR	value	Specifies the font color for navigator links.
LINK_FONT_BOLD	ON OFF	Specifies whether the font is bold faced for navigator links.
LINK_FONT_ITALIC	ON OFF	Specifies whether the font is italicized for navigator links.
LINK_FONT_UNDERLINE	ON OFF	Specifies whether the font is underlined for navigator links.
LINK_FONT_STRIKEOUT	ON OFF	Specifies whether the font for navigator link is displayed with a horizontal line through it.
LINK_IMG_FILE_FIRST	value	Specifies the hyperlink's image file for first page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_PREV	value	Specifies the hyperlink's image file for previous page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_NEXT	value	Specifies the hyperlink's image file for next page when USE_TEXT_LINKS option is set to OFF.
LINK_IMG_FILE_LAST	value	Specifies the hyperlink's image file for last page when USE_TEXT_LINKS option is set to OFF.
NAV_TYPE	PAGE SCREEN	Specifies the type of the navigator. If NAV_TYPE is set to SCREEN then the navigator will be fixed to screen, else the navigator will be fixed to page. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_ORIENT	VERTICAL HORIZONTAL	Specifies the orientation of the navigator. If NAV_ORIENT is set to HORIZONTAL then, a Horizontal navigator is encoded, else a Vertical navigator is encoded. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
NAV_POS	TOPLEFT TOPCENTER TOPRIGHT CENTERLEFT CENTER CENTERRIGHT BOTTOMLEFT BOTTOMCENTER BOTTOMRIGHT	Specifies the position of the navigator. The property is effective only when the options for SEP_FILE_PER_PAGE and NAV_SHOW are set to ON.
DISABLE_RIGHT_CLICK	ON OFF	Specifies whether to disable the typical speed menu options when you right click on your mouse button.
DISABLE_COPYING	ON OFF	Specifies whether to disable the option to highlight and copy the text.

AUTO_MAXIMIZE_WINDOW	ON OFF	Specifies whether to automatically maximize the window when HTML document is opened.
----------------------	-----------	--

Note: Each additional OPTION parameter must be separated by the pipe | symbol.

Example:

```
--To print SalesSumAndSubTot report as XHTML file
PRINT SalesSumAndSubTot +
OPTION XHTML +
|FILENAME SalesSumAndSubTot.htm +
|SHOW_CANCEL_DIALOG ON +
|BACKGROUND_FILE NONE +
|BACKGROUND_TYPE CENTER +
|BACKGROUND_COLOR WHITE +
|INCLUDE_LINES ON +
|INCLUDE_SHAPES ON +
|INCLUDE_RICHTEXT OFF +
|RICHTEXT_ENCODING_TYPE PLAINTEXT +
|INCLUDE_IMAGES ON +
|IMAGE_FORMAT JPG +
|JPEG_QUALITY 100 +
|IMAGE_DPI -1 +
|PIXELFORMAT 24 +
|INCLUDE_HYPERLINKS ON +
|TITLE Customer Sales Sub-Totals and Totals +
|SUBJECT ConComp Sales Summary Report +
|AUTHOR Accounting Team of ConComp +
|KEYWORDS SubTotals Totals Sales Summary +
|PAGE_END_LINES ON +
|OUTPUT_STYLES_TO_CSS_FILE ON +
|CSSFILE +
|OUTPUT_SCRIPTS_TO_JAVA_SCRIPT_FILE ON +
|JAVA_SCRIPT_FILE_NAME +
|GENERATE_TOC ON +
|TOC_FILE_NAME Index.htm +
|IMAGE_FOLDER Images +
|DEFAULT_FONT_NAME Verdana +
|DEFAULT_FONT_SIZE 9 +
|DEFAULT_FONT_COLOR BLACK +
|DEFAULT_FONT_BOLD OFF +
|DEFAULT_FONT_ITALIC OFF +
|DEFAULT_FONT_UNDERLINE OFF +
|DEFAULT_FONT_STRIKEOUT OFF +
|SEP_FILE_PER_PAGE ON +
|NAV_SHOW ON +
|LINK_BACK_COLOR WHITE +
|LINK_H_BACK_COLOR BLUE +
|LINK_H_FORE_COLOR WHITE +
|USE_TEXT_LINKS OFF +
|LINK_TEXT_FIRST First +
|LINK_TEXT_PREV Previous +
|LINK_TEXT_NEXT Next +
|LINK_TEXT_LAST Last +
|LINK_FONT_NAME Wingdings +
|LINK_FONT_SIZE 10 +
```

```

|LINK_FONT_COLOR BLUE +
|LINK_FONT_BOLD OFF +
|LINK_FONT_ITALIC OFF +
|LINK_FONT_UNDERLINE OFF +
|LINK_FONT_STRIKEOUT OFF +
|LINK_IMG_FILE_FIRST Images\FirstOff.gif +
|LINK_IMG_FILE_PREV Images\PrevOff.gif +
|LINK_IMG_FILE_NEXT Images\NextOff.gif +
|LINK_IMG_FILE_LAST Images\LastOff.gif +
|NAV_TYPE PAGE +
|NAV_ORIENT HORIZONTAL +
|NAV_POS BOTTOMCENTER +
|DISABLE_RIGHT_CLICK OFF +
|DISABLE_COPYING OFF +
|AUTO_MAXIMIZE_WINDOW ON +
|OPEN ON

```

1.17.6.21 XLS

The XLS output option displays the report in a Microsoft Excel format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLS
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ALL_TEXT_IN_GENERAL_FORMAT	ON OFF	Specifies whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly (Integer, Double etc).
CELL_ATTR	ON OFF	Cell attributes include shading of the cell whose color is not WHITE.
INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
IMAGE_FORMAT	BMP WMF	Specifies the image file format for image controls used in the report.
INCLUDE_LINES	ON OFF (default)	Specifies whether lines on the report should be included in the exported file.
INCLUDE_RICH_TEXT	ON (default) OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TYPE	IMAGE (default) PLAINTEXT	Indicates the way in which Rich Text contents should be encoded in the output. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode RichText contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode RichText contents as unformatted text. The font used when encoding RichText as plain text is the one set in the Font property of the RichText object. When

		<p>RICHTEXT_ENCODING_TYPE is set to PLAINTEXT make sure you set the Default Font property of the RichText control in report designer to true and set the required font in the Font property.</p>
INCLUDE_SHAPES	ON OFF (default)	Specifies whether shapes on the report should be included in the exported file.
LINE_SPACE	ACTUAL 0 1 2 3 4 5	Specifies the spacing to be used between lines of text.
ONE_SHEET_PER_PAGE	ON OFF (default)	Specifies whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.
MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.

- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the report data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Examples:

--To print Invoices as XLS file

```
PRINT Invoice WHERE CustID = 127 +
OPTION XLS +
|FILENAME Invoice.XLS +
|SHOW_CANCEL_DIALOG OFF +
|ALL_TEXT_IN_GENERAL_FORMAT OFF +
|CELL_ATTR ON +
|INCLUDE_IMAGES OFF +
|INCLUDE_LINES OFF +
|INCLUDE_RICH_TEXT ON +
|RICHTEXT_ENCODING_TYPE IMAGE +
|INCLUDE_SHAPES OFF +
|LINE_SPACE ACTUAL +
|ONE_SHEET_PER_PAGE ON +
|OPEN ON
```

--To print an encrypted spreadsheet

```
PRINT BonusRates OPTION XLS +
|FILENAME BonusRate.xls +
|PASSWORD PeachCobbler
```

1.17.6.22 XLS2003

The XLS2003 output option displays the report in a Microsoft Excel 2003 format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLS2003
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ALL_TEXT_IN_GENERAL_FORMAT	ON OFF	Specifies whether all the text in the report should be converted to general format or the format of the text should be detected and encoded accordingly (Integer, Double etc).
CELL_ATTR	ON OFF	Cell attributes include shading of the cell whose color is not WHITE.
INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
IMAGE_FORMAT	BMP WMF	Specifies the image file format for image controls used in the report.
INCLUDE_LINES	ON OFF (default)	Specifies whether lines on the report should be included in the exported file.

INCLUDE_RICH_TEXT	ON (default) OFF	Specifies whether contents of the Rich Text objects on the report should be included in the exported file.
RICHTEXT_ENCODING_TYPE	IMAGE (default) PLAINTEXT	Indicates the way in which Rich Text contents should be encoded in the output. Set RICHTEXT_ENCODING_TYPE to IMAGE to encode RichText contents as exact graphical representations. Set RICHTEXT_ENCODING_TYPE to PLAINTEXT to encode RichText contents as unformatted text. The font used when encoding RichText as plain text is the one set in the Font property of the RichText object. When RICHTEXT_ENCODING_TYPE is set to PLAINTEXT make sure you set the Default Font property of the RichText control in report designer to true and set the required font in the Font property.
INCLUDE_SHAPES	ON OFF (default)	Specifies whether shapes on the report should be included in the exported file.
LINE_SPACE	ACTUAL 0 1 2 3 4 5	Specifies the spacing to be used between lines of text.
ONE_SHEET_PER_PAGE	ON OFF (default)	Specifies whether each page in the report should be encoded into separate excel sheets or all the pages should be encoded in a single excel sheet.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
PAGE_SETTINGS	ALL FIRST LAST PAGE_LIST	Specifies the option to print All, First, Last or a given number of pages. When PAGE_LIST is set as the value for PAGE_SETTINGS, the PAGE_LIST_VALUES parameter must be used with option in order to specify the page list.
PAGE_LIST_VALUES	value	Specifies the values for number of pages, such as, 3,5,8 or 3,8 or 3, or 1-3,8 or 3,5,9-15. This option must be used with the PAGE_SETTINGS parameter when PAGE_LIST is specified.
PAGE_STYLE_SETTINGS	ALL FIRST LAST Page List e.g. 1,4-6,10-12,20	Determines the pages in which the Page Style will appear on.
MARGIN_LEFT	value	Specifies the horizontal position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_LEFT parameter is set to 0.25 inches and you place a report component in a band and set the component's "left" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the left margin).
MARGIN_TOP	value	Specifies the vertical position on the page where printing should begin. All report component positions are relative to the margin. In other words, if the MARGIN_TOP parameter is set to 0.25 inches and you place a report component in a band and set the component's "top" parameter to 0, then that component will print 0.25 inches from the edge of page (or at the top margin).
MARGIN_RIGHT	value	Specifies the horizontal position on the page where printing should stop. All report component positions are relative to the margin.

MARGIN_BOTTOM	value	Specifies the vertical position on the page where printing should stop. All report component positions are relative to the margin.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.
- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the report data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Examples:

```
--To print Invoices as XLS file
PRINT Invoice WHERE CustID = 127 +
OPTION XLS2003 +
|FILENAME Invoice.XLS +
|SHOW_CANCEL_DIALOG OFF +
|ALL_TEXT_IN_GENERAL_FORMAT OFF +
|CELL_ATTR ON +
|INCLUDE_IMAGES OFF +
|INCLUDE_LINES OFF +
|INCLUDE_RICH_TEXT ON +
|RICHTEXT_ENCODING_TYPE IMAGE +
|INCLUDE_SHAPES OFF +
|LINE_SPACE ACTUAL +
|ONE_SHEET_PER_PAGE ON +
|OPEN ON
```

```
--To print an encrypted spreadsheet
PRINT BonusRates OPTION XLS2003 +
|FILENAME BonusRate.xls +
|PASSWORD PeachCobbler
```

1.17.6.23 XLSX

The XLSX output option displays the report in a Microsoft Excel Workbook format file.

Syntax:

```
PRINT reportname ARRANGE clause WHERE clause .. ORDER BY clause .. OPTION XLSX
```

The following OPTION parameters are available to customize the output:

Parameter	Value	Description
FILENAME	value	Specifies the full path and file name. If no path is defined, output will be saved in the current working directory.
PASSWORD	value	Specifies the password for the Excel file. Passwords are case sensitive.
SHOW_CANCEL_DIALOG	ON (default) OFF	Specifies whether to suppress cancel dialog box during the print process.
ROW_SPACING	value	Specifies the spacing between spreadsheet rows.
COLUMN_SPACING	value	Specifies the spacing between spreadsheet columns.

INCLUDE_IMAGES	ON OFF	Specifies whether image objects on the report should be included in the exported file.
INCLUDE_TEXT	ON OFF	Specifies whether text is included in the exported file.
PAGES_PER_WORKSHEET	ON OFF	Specifies the page limit per worksheet.
PRINT_GRID_LINES	ON OFF	Specifies whether grid lines are displayed in the exported file.
PASS_SETTINGS	value	Specifies the number of times a report traverses the data before generating pages.
OPEN	ON OFF	Specifies whether the report should be opened after it is generated.
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment. See additional options when EMAIL value is set to ON. EMAIL SET ON

Notes:

- Each additional OPTION parameter must be separated by the pipe | symbol.
- When printing to the XLS format, the report objects must be spaced apart appropriately or "padded" in order for the report data to appear correctly in the Excel report output. Experimenting with vertical and horizontal will also allow you to attain your desired output. Also, please refrain from using elaborate fonts that may not be supported within Excel.

Examples:

```
--To print Invoices as an XLSX file
PRINT Invoice WHERE CustID = 127 +
OPTION XLSX +
|FILENAME Invoice.XLSX +
|SHOW_CANCEL_DIALOG OFF +
|ROW_SPACING 1 +
|COLUMN_SPACING 1 +
|PRINT_GRID_LINES ON +
|OPEN ON
```

```
--To print an encrypted spreadsheet
PRINT BonusRates OPTION XLSX +
|FILENAME BonusRate.xlsx +
|PASSWORD PeachCobbler
```

1.17.6.24 EMAIL Parameter

The following OPTION parameters are available when the **EMAIL** value is set to **ON**. When using EMAIL ON, the printed report document is automatically attached to the email. The EMAIL_ATTACHMENTS parameter is for other external file attachments.

Parameter	Value	Description
EMAIL	ON OFF	Specifies the report output (defined as FILENAME filename.ext) to be sent via e-mail as an attachment.
EMAIL_AUTHENTICATION	ON OFF	Specifies if the SMTP Server requires user authentication.
EMAIL_HOST	value	Specifies Hostname/IP Address of SMTP Server (Outgoing). Also sometimes known as your 'Sending Mail Server', your SMTP server receives and processes all of your outgoing mail. For example, your Outgoing Mail Server might be called something like 'my-internet-provider.com' or 'outgoing-mail.mycompany.com'. The SMTP server name is

		sometimes the same as your incoming mail server name. If your ISP or your network administrator has not given this server name to you, you can leave the box blank. If you do not know what your outgoing mail server name is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_PORT	value	Specifies the integer value of SMTP Server Port. (Default = 25).
EMAIL_USERID	value	Specifies a valid User Name or Login Name for SMTP Server. This is the name you use to log into your email account. Your ISP or organization provides it. It is generally, but not necessarily, the name that appears before the at sign (@) in your return email address. For example, if your email address was <jim@my-server.com> - then your username would be 'jim' (no quotes). If you do not know what your username is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_PASSWORD	value	Specifies a valid SMTP Server Password for the user.
EMAIL_ATTACHMENTS	value	Specifies the attached file list. Each file must be separated by the pound character (#).
EMAIL_BODY	value	Specifies the body of the mail.
EMAIL_DATE	value	Specifies the date the message was sent. (Default = .#DATE)
EMAIL_FROM_ADDRESS	value	Specifies the address of the E-mail account from which the mail is to be sent. Sender E-Mail Address (From E-Mail Address:) Enter your e-mail address here. For example, your email address might be 'jim@my-internet-service.com' or 'molly@my-business-domain.com' Take care not to make a mistake here - or else it will make it difficult for others to reply to your emails. If you do not know what your e-mail address is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_FROM_NAME	value	Specifies the person's name in an E-mail address. Your real name or what you want to appear as your name on messages that you send out.
EMAIL_TO_LIST	value	Specifies the recipient(s) of a message. Specify the list of valid e-mail address(s) to which the e-mail is to be sent.
EMAIL_REPLY	value	Specifies the delivery address for message replies. This is typically the same as Sender E-Mail Address. Enter your e-mail address here. For example, your email address might be 'jim@my-internet-service.com' or 'molly@my-business-domain.com' Take care not to make a mistake here - or else it will make it difficult for others to reply to your emails. If you do not know what your e-mail address is, you need to contact your Internet Service Provider or Network Administrator.
EMAIL_SUBJECT	value	Specifies the subject for the message. Subject is a text string that identifies the subject for the message.
EMAIL_CC_LIST	value	Specifies the list of e-mail addresses as Carbon Copies to be sent along with the e-mail.
EMAIL_BCC_LIST	value	Specifies the list of e-mail addresses as Blind Carbon Copies to be sent along with the mail.
EMAIL_DELETE_AFTER_SEND	ON OFF	Specifies if the report files are deleted after sending the e-mail.
EMAIL_SHOW_DIALOG	ON OFF	Specifies if the e-mail setup dialog is displayed. The RBTI_PRN_EMAIL_DLG variable will be defined after the dialog is displayed and is set with "OK" when the OK button is selected. Otherwise, it is set as "CANCEL".

Note: Each additional OPTION parameter must be separated by pipe | character.

Examples:**--Multiple Attachment Example:**

```

PRINT Invoice WHERE TransID = 1031 +
OPTION PDF|FILENAME Invoice_1031.PDF +
|EMAIL ON +
|EMAIL_TO_LIST yourname@yourdomain.com, whoelse@yourdomain.com +
|EMAIL_FROM_NAME John D. Johnson +
|EMAIL_FROM_ADDRESS johndjohnson@yourdomain.com +
|EMAIL_SUBJECT Imagine The Possibilities! +
|EMAIL_BODY Attached is the invoice - sent via R:BASE. Because we Can! +
|EMAIL_ATTACHMENTS C:\Letters\PrivacyPolicy.pdf#C:\Letters\PaymentTermsPolicy.pdf +
|EMAIL_DELETE_AFTER_SEND ON +
|EMAIL_SHOW_DIALOG OFF

```

--Multi-line Message Body Example:

```

SET VAR vEmailBody TEXT = +
('Attached please find your sales summary report!'+(CHAR(013)))+ +
(CHAR(013))+'Please reply with any questions.')
PRINT InvoiceSummary WHERE CustID = 128 +
OPTION PDF|FILENAME C:\InvoiceSummary_123.PDF +
|EMAIL ON +
|EMAIL_TO_LIST MMurdock@NelsonMurdockLawOffice.com,FNelson@NelsonMurdockLawOffice.com
+
|EMAIL_FROM_NAME Wilbur Day +
|EMAIL_FROM_ADDRESS Wilbur.Day@KaxtonLabs.com +
|EMAIL_SUBJECT Invoice Summary As Requested +
|EMAIL_BODY .vEmailBody +
|EMAIL_DELETE_AFTER_SEND ON +
|EMAIL_SHOW_DIALOG OFF

```

1.17.7 PRNSETUP

Use the PRNSETUP command to open the "Printer Setup" dialog box, and control the printer options. If you wish to use more than one option you will need to issue these as separate commands. If you change the printer using the 'PRINTER NAME' parameter, the other printer options will be reset. So, be sure to issue the 'PRINTER NAME' parameter first.

PRNSETUP

' PRINTER NAME'

' PAPER SIZE'

' PAPER SOURCE'

COLLATE value

COLOR value

COPY n

DUPLEX value

FONTS

HELP

INFO

LANDSCAPE

OPTION

CAPTION 'message'

LEFT value

STANDARD

THEMENAME value

TOP value

QUALITY value

PAPER NAMES

PORTRAIT

PRINTERS

TRAYS

TRAY_NAME value

Print Setup

Printer

Name:

Microsoft Print to PDF

Properties...

Status:

Ready

Type:

Microsoft Print To PDF

Where:

PORTPROMPT:

Comment:


Paper

Size:


Letter

Source:

Orientation



☒ Portrait



☐ Landscape

Network...

OK

Cancel

Options

'PAPER SIZE'
Specifies the paper size. Options:

LETTER	Letter 8 12 x 11 in
LETTERSMALL	Letter Small 8 12 x 11 in
TABLOID	Tabloid 11 x 17 in
LEDGER	Ledger 17 x 11 in
LEGAL	Legal 8 12 x 14 in
STATEMENT	Statement 5 12 x 8 12 in
EXECUTIVE	Executive 7 14 x 10 12 in

A3	A3 297 x 420 mm
A4	A4 210 x 297 mm
A4SMALL	A4 Small 210 x 297 mm
A5	A5 148 x 210 mm
B4	B4 (JIS) 250 x 354
B5	B5 (JIS) 182 x 257 mm
FOLIO	Folio 8 12 x 13 in
QUARTO	Quarto 215 x 275 mm
10X14	10x14 in
11X17	11x17 in
NOTE	Note 8 12 x 11 in
ENV9	Envelope #9 3 78 x 8 78
ENV10	Envelope #10 4 18 x 9 12
ENV11	Envelope #11 4 12 x 10 38
ENV12	Envelope #12 4 276 x 11
ENV14	Envelope #14 5 x 11 12
CSHEET	C size sheet
DSHEET	D size sheet
ESHEET	E size sheet
ENVDL	Envelope DL 110 x 220mm
ENVC3	Envelope C3 324 x 458 mm
ENVC4	Envelope C4 229 x 324 mm
ENVC5	Envelope C5 162 x 229 mm
ENVC6	Envelope C6 114 x 162 mm
ENVC65	Envelope C65 114 x 229 mm
ENVB4	Envelope B4 250 x 353 mm
ENVB5	Envelope B5 176 x 250 mm
ENVB6	Envelope B6 176 x 125 mm
ENVITALY	Envelope 110 x 230 mm
ENVMONARCH	Envelope Monarch 3.875 x 9.5 in
ENVPERSONAL	6 34 Envelope 3 58 x 6 12 in
FANFOLDUS	US Std Fanfold 14 78 x 11 in
FANFOLDSG	German Std Fanfold 8 12 x 12 in
FANFOLDLG	German Legal Fanfold 8 12 x 13 in
ISOB4	B4 (ISO) 250 x 353 mm
JPPOSTCARD	Japanese Postcard 100 x 148 mm
9X11	9 x 11 in
10X11	10 x 11 in
15X11	15 x 11 in
ENVINVITE	Envelope Invite 220 x 220 mm
LETTEREXTRA	Letter Extra 9 275 x 12 in
LEGALEXTRA	Legal Extra 9 275 x 15 in
TABLOIDEXTRA	Tabloid Extra 11.69 x 18 in
A4EXTRA	A4 Extra 9.27 x 12.69 in
LETTERTRANSVERSE	Letter Transverse 8 275 x 11 in
A4TRANSVERSE	A4 Transverse 210 x 297 mm
LETTEREXTRATRANSVERSE	Letter Extra Transverse 9 275
APLUS	SuperASuperAA4 227 x 356 mm
BPLUS	SuperBSuperBA3 305 x 487 mm
LETTER_PLUS	Letter Plus 8.5 x 12.69 in
A4PLUS	A4 Plus 210 x 330 mm
A5TRANSVERSE	A5 Transverse 148 x 210 mm
B5TRANSVERSE	B5 (JIS) Transverse 182 x 257 mm
A3EXTRA	A3 Extra 322 x 445 mm
A5EXTRA	A5 Extra 174 x 235 mm
B5EXTRA	B5 (ISO) Extra 201 x 276 mm
A2	A2 420 x 594 mm
A3TRANSVERSE	A3 Transverse 297 x 420 mm
A3EXTRATRANSVERSE	A3 Extra Transverse 322 x 445 mm

'PAPER SOURCE'

Specifies the paper source. Options:

- UPPER
- LOWER
- MIDDLE
- MANUAL
- ENVELOPE
- ENVMANUAL
- AUTO
- TRACTOR
- SMALLFMT
- LARGEfmt
- LARGEcapacity
- CASSETTE

'PRINTER NAME'

Specifies the printer name to print to. This must match the name of the printer as it is displayed in the Windows Printer Control Panel.

COLLATE value

Specifies whether collation should be used when printing multiple copies. Options:

- ON - Collate when printing multiple copies.
- OFF - Do not collate when printing multiple copies.

COLOR value

Switches between color and monochrome on color printers. Options:

- ON
- OFF

COPY n

Specifies the number of copies to print.

DUPLEX

Select duplex or double-sided printing for printers capable of duplex printing. Horizontal duplex printing is performed across the page, from left to right, so the text will read as in a book. Vertical duplex printing is performed down the page, from top to bottom, so the text will read as a tablet. Options:

- SIMPLEX - Normal (non-duplex) printing
- HORIZONTAL - Long-edge binding, where the top of both sides of the page are at the same end of the sheet
- VERTICAL - Short-edge binding, where the bottom of one page is at the same end of the sheet as the top of the next page

Fonts

Reads installed Fonts and returns a list of the fonts supported by the printer.

HELP

Displays available options to retrieve additional information.

INFO

Retrieves a list of current printer settings like:

- Current Printer :
- Paper Orientation :
- Paper Size :
- Paper Source :
- Print Quality (DPI) :
- Copies :
- Color Mode :
- Duplex Mode :
- Collation :

Note: Only use options specific to your printer.

LANDSCAPE

Specifies the landscape orientation for the printer output.

OPTION parameters - provides the ability to customize the Print Setup window. More than one OPTION parameter may be specified, each must be separated with the pipe character (|).

Parameters	Value	Description
CAPTION	value	specifies the text of the caption message to display in the window. The value can be passed as a variable.
LEFT	nn	specifies the print dialog location (in pixels) from the left side of the screen
STANDARD		displays the standard Windows operating system Printer Setup dialog window. The dialog window supports all other available option except the use of themes. The available printer options will vary on different operating systems.
THEMENAME	value	displays a theme for the dialog window
TOP	nn	specifies the print dialog location in pixels from the top of the screen

QUALITY

Specifies the printer resolution. Options:

- DRAFT
- HIGH
- LOW
- MEDIUM

PAPER NAMES

Retrieves a list of supported paper names (Letter, Legal, etc.).

PORTRAIT

Specifies the portrait orientation for the printer output.

PRINTERS

Lists all installed printers.

TRAY_NAME

Specifies the paper tray bin for the printer output. It is recommended to use the actual tray bin names as seen in the printer setup dialog. Otherwise, the following constants may be specified: UPPER, LOWER, MIDDLE, MANUAL, ENVELOPE, ENVMANUAL, AUTO, TRACTOR, SMALLFMT, LARGE FMT, LARGE CAPACITY, CASSETTE

TRAYS

Retrieves a list of the paper bin names.

Notes:

- Specifying an invalid printer name will not result in an error message, and the current printer will not be changed.
- The PRNSETUP dialog will return the RBTI_PRNSETUP system variable to capture the user's button action/selection. The [OK] button will return the value of "OK" for the RBTI_PRNSETUP variable. The [Cancel] button or closing the PRNSETUP dialog will return the value of "CANCEL" for the RBTI_PRNSETUP variable.
- Use the PRNSETUP INFO command for detailed information on the current printer.

Examples:

```
-- Example 01 (Default Window):
PRNSETUP
```

-- Example 02 (Changes the printer and then shows the window):

```
PRNSETUP 'Adobe PDF'
PRNSETUP
```

-- Example 03 (Custom caption):

```
PRNSETUP OPTION CAPTION 'Select Printer'
```

-- Example 04 (Custom caption with the Vista CG theme):

```
PRNSETUP OPTION CAPTION 'Select Printer' |THEMENAME Vista CG
```

-- Example 05 (Custom caption with the dialog set to appear at a specific screen location):

```
PRNSETUP OPTION CAPTION 'Select the Printer' |TOP 150 |LEFT 150
```

1.17.8 PROCESSMESSAGE

Use the PROCESSMESSAGE command to process messages that are currently in the windows message queue.

PROCESSMESSAGE

It is helpful to show a progress message when performing long running tasks. When processing a time-consuming routine, there are ways to let the user know the status of the process using the [PAUSE](#) 3 with [GAUGE](#) options as well as the use of PROCESSMESSAGE command, which processes messages that are currently in the windows message queue. The PROCESSMESSAGE can help in the GUI part to avoid the "Not responding" behavior in Windows operating systems. A common use of PROCESSMESSAGE is in long [WHILE](#) loops.

PROCESSMESSAGE may be called in each loop iteration to give the GUI time to process the pending Windows messages. For a loop that only does data processing, PROCESSMESSAGE can also be used. It is advised to disable GUI update settings like [UINOTIF](#) before entering the loop with PROCESSMESSAGE, to counter some side-effects of PROCESSMESSAGE.

It is also important to not overuse PROCESSMESSAGE. Use the command only in places where it is necessary for the GUI to "breathe" during a long running task.

PROCESSMESSAGE can be called after every iteration in the WHILE loop cycle. The following example demonstrates where to best place the PROCESSMESSAGE command.

When using the PAUSE command, the [NO_FOCUS](#) option can also be included in the OPTION parameters, so the dialog will not be focused when displayed. The no focus feature will benefit to possibly prevent an interruption in the focus transition in an active form.

Examples:

```
--Example 01 - PAUSE with Moving GIF:
SET VAR vPauseMessage TEXT = ((CHAR(013))+'Computing - Please Wait .....')
CLS
PAUSE 3 USING .vPauseMessage +
CAPTION ' Pause 3 with PROCESSMESSAGE' +
OPTION ICON_FILE Hourglass.GIF +
|BACK_COLOR WHITE +
|MESSAGE_COLOR WHITE +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_COLOR RED +
|MESSAGE_FONT_SIZE 10 +
|THEMENAME R:BASE Rocks!
SET VAR vCounter INT = 1
WHILE vCounter < 2500000 THEN
```

```
SET VAR vCounter = (.vCounter + 1)
PROCESSMESSAGE
ENDWHILE
CLEAR VARIABLE vPauseMessage,vCounter
CLS
RETURN
```

--Example 02 - PAUSE with GAUGE:

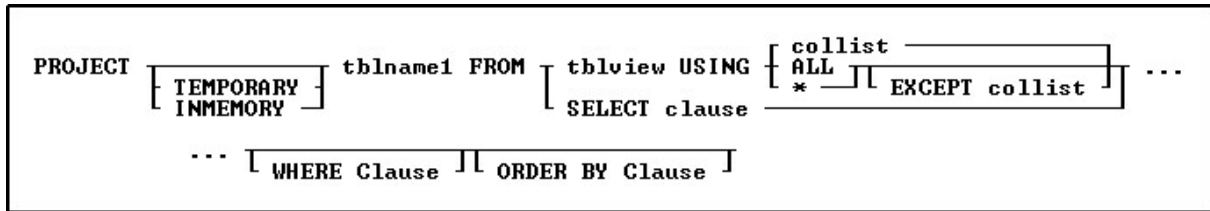
```
CLS
PAUSE 3 USING ' Calculating ... Please Stand By ...' +
CAPTION ' Pause 3 with Gauge + PROCESSMESSAGE' ICON APP +
OPTION GAUGE_VISIBLE ON +
|GAUGE_COLOR [R218,G228,B246] +
|GAUGE_INTERVAL 10 +
|MESSAGE_FONT_NAME VERDANA +
|MESSAGE_FONT_SIZE 10 +
|MESSAGE_FONT_COLOR BLUE +
|THEMENAME Razzmatazz
SET VAR vCounter INTEGER = 1
WHILE vCounter < 2500000 THEN
SET VAR vCounter = (.vCounter + 1)
PROCESSMESSAGE
ENDWHILE
CLEAR VARIABLE vCounter
CLS
RETURN
```

--Example 03 - PAUSE with GAUGE and NO_FOCUS:

```
CLS
PAUSE 3 USING 'Collecting data ... this will take a few seconds ...' +
CAPTION 'Using Pause with Gauge' ICON WINDOWS +
OPTION BACK_COLOR WHITE +
|MESSAGE_FONT_NAME Tahoma +
|MESSAGE_FONT_COLOR NAVY +
|MESSAGE_FONT_SIZE 12 +
|GAUGE_VISIBLE ON +
|GAUGE_COLOR GRAY +
|GAUGE_INTERVAL 10 +
|NO_FOCUS
SET VAR vCounter INTEGER = 1
WHILE vCounter < 250000 THEN
SET VAR vCounter = (.vCounter + 1)
PROCESSMESSAGE
ENDWHILE
CLEAR VARIABLE vCounter
CLS
RETURN
```


1.17.9 PROJECT

Use the PROJECT command to create a new table from an existing table or view.



Options

Specifies to use all columns with the command.

ALL

Specifies to use all columns with the command.

collist

Specifies the column(s) to use with the command.

EXCEPT

Specifies the column(s) which will not be included in the projected table.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using INMEMORY in place of TEMPORARY creates the temporary table in memory, rather than a temporary file.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

SELECT clause

Specifies the columns and one or more tables or views from which to create the new table. Using the [SELECT](#) portion it is not necessary to create a view first in order to perform PROJECT into a new table from multiple table joins. A USING clause is not needed as all required columns are defined in the SELECT statement.

tblname1 FROM tblview

tblname1 is the name of the new table you want to create, and FROM *tblview* specifies the existing table or view you want to copy.

TEMPORARY

Allows you to create a Temporary Table with the PROJECT command.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the PROJECT Command

The new table can be a copy of an existing table, a copy of an existing table with the rows sorted in a different order, a duplicate of a table structure without any data, a table that contains specific rows and columns from an existing table, or a SELECT clause which specifies the columns and one or more tables or views from which to create the new table.

When not using SELECT, you must include the USING clause with the PROJECT command. The USING clause specifies the columns to be included in the new table. If you want the new table to include all the columns from an existing table, use an asterisk (*) in the clause. If you want the new table to include only specific columns from the existing table, list them in the order you want them to appear in the new table. If you want the new table to include all columns in a different order, list them in the order you want them to appear.

Working with Computed Columns

R:BASE copies the data from each column into the new table. If a computed column is included, R:BASE transfers the current values in the computed column to the new table. In order to calculate computed values in the column in the new table, R:BASE needs the column names used in the computed column's expression. Therefore, include those column names in the USING clause before the computed column. When you do not include those column names in the USING clause before the computed column, R:BASE makes the computed column a regular column, assigns a data type, and displays a message suggesting you rename the column in the new table. If you do not rename the column, the new table has a column with the same name as the column in the original table, but does not have the designation COMPUTED. You will not be able to use the [BACKUP](#) and [UNLOAD](#) commands, because you cannot have a computed column and a regular column with the same name.

Working with Autonumbered Columns

In a new table, R:BASE does not update the value in a row for the autonumber column. The autonumber column becomes a regular column.

Removing Columns and Rows from a Table

PROJECT is also useful if you want to remove several columns or rows from a table. To delete columns from a table, create a new table that retains the columns you want to keep, or to delete rows, create a new table using a [WHERE](#) clause. Use the [DROP](#) command to remove the table you no longer want, then use the [RENAME](#) command to give the new table the original table's name.

Transferring Default Column Definitions

Default column definitions are transferred to a new table. If you want to change the projected default column definition, define the desired default column definition with the [ALTER TABLE](#) command.

Examples

The following command creates a new table that is a duplicate of the *employee* table.

```
PROJECT Reps FROM Employee USING *
```

The following command creates a table named *Empty* that has the same structure as the *ProdLocation* table but contains no rows of data.

```
PROJECT Empty FROM ProdLocation USING * WHERE LIMIT = 0
```

The example below creates a table named *EmpHires*. The order of the columns in the *EmpHires* table are specified in the USING clause. The [WHERE](#) clause specifies that only the information for employees hired before January 1, 2020 will be selected. The [ORDER BY](#) clause sorts the rows in alphabetical order by the employees' last names.

```
PROJECT EmpHires FROM Employee USING EmpFName, EmpLName, +
EmpID, EmpExt, HireDate WHERE HireDate < '01/01/2020' +
ORDER BY EmpLName
```

The following creates a new table from the *Staff* and *Departments* table, with the SELECT statement to specify the column and table source.

```
PROJECT StaffDepts FROM +
SELECT T2.DepartmentID,T2.Description,T2.OwnerDept,+
T1.LastName,T1.FirstName,T1.PhoneExt,T2.DeptShape +
FROM Staff T1,Departments T2 +
WHERE T1.DepartmentID = T2.DepartmentID
```

The following creates a temporary table *CustDetails* with all *Customer* table columns except *CustFax*.

```
PROJECT TEMPORARY CustDetail FROM Customer USING * EXCEPT CustFax
```

1.17.10 PROPERTY

Use the PROPERTY command to dynamically change the properties of form control object(s), the form itself, form tables, the R:BASE application, themes, and more.

PROPERTY	ComponentID	ControlProperty	Value
	RBASE_FORM	FormProperty	Value
	RBASE_FORM_ACTION	CustomFormActionName	Parameters
	TABLE	FormTableName	Value
	TABLE_COLUMN	ColumnProperty	Value
	APPLICATION	ApplicationProperty	Value
	RBA_FORM	ApplicationFormProperty	Value
	REPORT	ReportProperty	Value
	BLOB_EDITOR	BlobEditorProperty	Value
	CLIPBOARD	ClipboardProperty	Value
	LOADWINDOW	WindowProperty	Value
	ThemeParameter	ThemeName	Value
	ROSK	RBASEOnScreenKeyboardProperty	Value
	SPEECH	SpeechProperty	Value

Options

APPLICATION ApplicationProperty

Specifies to alter an application property such as the CAPS lock status, if it is compiled, or the title

BLOB_EDITOR BlobEditorProperty

Specifies to alter the display of the BLOB Editor utility

CLIPBOARD ClipBoardProperty

Specifies to utilize the Windows clipboard

Component ID

Specifies the unique identifier assigned to a control (e.g. DB Edit). The Component ID within the PROPERTY command must match the unique identifier listed in the control properties. The "Component ID" field is located in the Object Properties for all forms, reports, labels, and controls. A "Component ID" option is also located in the speed menu list when you right click on a report/label control. Wildcards are supported for altering the properties of multiple components.

When using multiple forms, the **Component ID** portion of the PROPERTY command can consist two parts, the Component ID of the form, and the Component ID on a control on the form.

PROPERTY 'FormComponentID|ControlComponentID' Component_Property value

Example (Used to switch focus with MDI forms):

PROPERTY 'MDI_Customers|EGB_Next' SET_FOCUS 'TRUE'

ControlProperty

Specifies the name of the form/report/label object's property you wish to change

LOADWINDOW WindowProperty

To alter the load window property such as the caption or title. The LOADWINDOW parameter is only supported in R:Compiler.

Parameters

The optional value(s) you can pass onto the calling command, if required. Use a blank space ' ' with single or double quotes (depending on your database QUOTE settings), if parameter is not required in calling command block.

RBA_FORM ApplicationFormProperty

Specifies to alter an application form property such as the current theme, the enabled status for an action, or a caption

RBASE_FORM FormProperty

Specifies to alter a form property such as height, width, background, etc.

RBASE_FORM_ACTION CustomFormActionName

Specifies the actual name of the Command Name assigned when designing Custom Form Actions in the Form Designer

REPORT ReportProperty

Specifies to alter a report property such as the report name, the total page count, or the number of tables using in the report

ROSK RBASEOnScreenKeyboardProperty

Specifies to alter the display of the R:BASE On-Screen Keyboard (ROSK)

SPEECH SpeechProperty

Specifies to utilize the speech functionality for specified text

TABLE FormTableName

Specifies the driving table name of the form

TABLE_COLUMN ColumnProperty

Specifies the table's column value to alter

ThemeName

Specifies the external theme name to load, release, or change. For more information on using the PROPERTY command to manage external themes, see External Themes.

ThemeProperty

Specifies the theme property to manage

Value

Specifies the new value to assign to the property of the object specified

Tips:

- Properties of any control can be changed with assigned Component ID only.
- To change any property of a form control after the form is displayed, use the PROPERTY command(s) within the "On After Start EEP" section of the Form Properties.
- Always use the current [QUOTES](#) character around the *Value* parameter, if the value is text.

Notes:

- The complete list of PROPERTY command parameters are available within the **FormProperties.pdf** and **FormProperties_TOC.pdf** PDF documents, which are provided within the R:BASE program directory and are also available for download at the R:BASE Technologies Support page: <http://www.rbase.com/support/>
- An optional syntax builder add-on product called [R:Docs](#), containing all PROPERTY/GETPROPERTY commands and parameters constructed in an R:BASE application, is also available. Please contact the R:BASE Technologies Sales Team at sales@rbase.com if you wish to acquire an annual subscription to the stand-alone R:Docs application.

Examples:

Example 01.

To change the color of any control with Component ID:

```
PROPERTY ComponentID COLOR 'RED'
```

Example 02.

To hide the control:

```
PROPERTY ComponentID VISIBLE 'FALSE'
```

Example 03.

To change the font color:

```
PROPERTY ComponentID FONT_COLOR 'GREEN'
```

Example 04.

To change the font size:

```
PROPERTY ComponentID FONT_SIZE 36
```

Example 05.

To change the text alignment:

```
PROPERTY ComponentID ALIGN 'CENTER'
```

Example 06.

To change the form width:

```
PROPERTY RBASE_FORM WIDTH 300
```

Example 07.

To change the form height:

```
PROPERTY RBASE_FORM HEIGHT 300
```

Example 08.

To activate form Auto Scroll option:

```
PROPERTY RBASE_FORM AUTOSCROLL 'TRUE'
```

Example 09.

To change form color to teal:

```
PROPERTY RBASE_FORM COLOR 'TEAL'
```

Example 10.

To change form caption:

```
PROPERTY RBASE_FORM CAPTION 'New Form Caption'
```

Example 11.

To change form hint:

```
PROPERTY RBASE_FORM HINT 'New Form Hint'
```

Example 12.

To refresh table:

```
PROPERTY TABLE tablename 'REFRESH'
```

Example 13.

To move to the last record in the selected data set.

```
PROPERTY TABLE tablename 'LAST'
```

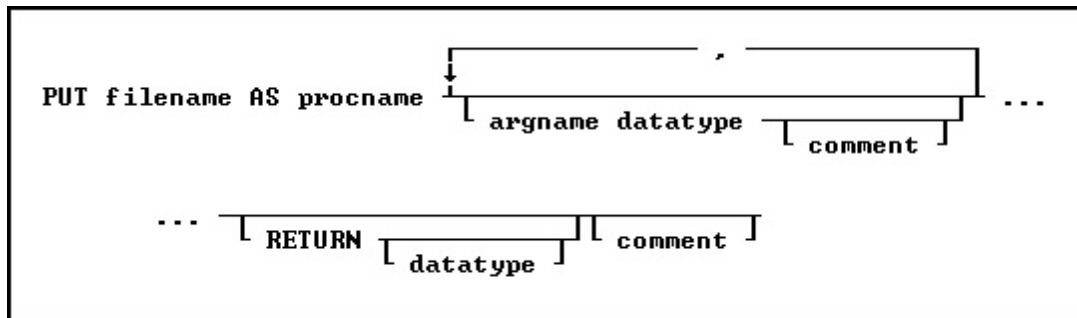
Example 14.

To move to the first record in the selected data set. Notice the quotes around the TEXT value while value with number without the quotes.

```
PROPERTY TABLE tablename 'FIRST'
```

1.17.11 PUT

Creates or replaces stored procedures into the database.



Options

argname datatype

The parameter name and data type. This portion may be repeated.

comment

An optional comment for the parameter or, if placed after RETURN, an optional comment for the entire procedure. The comment must be enclosed in the current quote setting.

filename

The filename in ASCII text format, with full path, to load as the stored procedure.

procname

Specifies the procedure name. If a procedure by this name already exists in the database, an error is generated. The procedure name is limited to 128 characters.

RETURN datatype

Determines the data type of the value returned by the procedure.

About the PUT Command

Argument List

When you load a stored procedure into a database, you specify arguments to be passed to it. These arguments are used within the procedure. When the procedure is called, the number and type of arguments passed must match the number and type specified when the procedure was stored in the database. When an argument name is referenced in the stored procedure code, the argument name must be preceded by a period unless it is a table or column name, then it must be preceded by an ampersand (&). For example:

```
UPDATE &p1 SET col = 99 WHERE col = .p2
```

The arguments names are specified when the procedure is stored in the database with the PUT command.

Return Values

The value to be returned by a stored procedure is specified in the procedure code following the keyword RETURN. For example, RETURN 'Los Angeles'. The value returned must match the datatype specified when the procedure was stored.

Replacing a Procedure

If you are replacing an existing procedure, you must LOCK the procedure first either with the [GET LOCK](#) or the [SET PROCEDURE](#) command. Once the procedure is locked, it is replaced by an updated file using the PUT command. A procedure cannot be replaced unless it is locked. A procedure is automatically unlocked when replaced with the PUT command.

Example

Use the PUT command as follows to store a command file as a stored procedure:

```
PUT INS.RMD AS SP_ContCheck p1 INT, p2 TEXT RETURN INTEGER
```

The contents of INS.RMD could be something like:

```
--INS.RMD
IF (.p1 > 105) THEN
  INSERT INTO contact (custid, contlname) VALUES (.p1, .p2)
  RETURN 1
ELSE
  RETURN 0
ENDIF
```

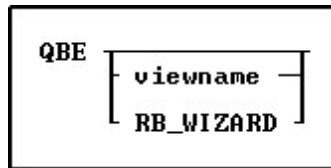
See Also:

Stored Procedures & Triggers

1.18 Q

1.18.1 QBE

Use the QBE command to launch the Query Builder, or the Query Wizard.



Options

viewname

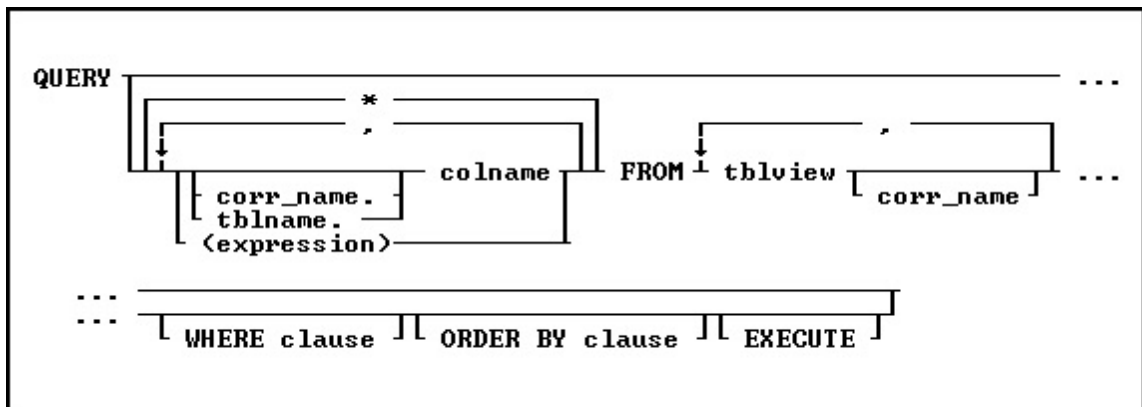
Specifies a view name.

RB_WIZARD

Launches the Query Wizard.

1.18.2 QUERY

Use the QUERY command to select, combine, manipulate, and retrieve data from tables.



Options

Specifies all columns.

,

Indicates that this part of the command is repeatable.

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter *#c*, where *#c* is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (*tblname.colname*). You can enter *tblname.** to specify all columns in the table.

corr_name

Correlation name. A nickname or alias for a table or view name. Use *corr_name* to refer to the same table twice within the command, or to more briefly specify a column in more than one table. The correlation name must be at least two characters.

EXECUTE

Retrieves columns and their data, then displays the columns with their data in the Data Browser.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

FROM tblview

Specifies the name of the table(s) or view(s) to query.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

tblname.

Specifies the table containing the columns. You can use *tblname.** to specify all columns in the table.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the QUERY Command

You construct a query to get information from your database's tables and columns. The process of selecting, retrieving, and working with data is referred to as creating and executing a query. R:BASE processes the query and displays what you request.

You can specify a maximum of ten tables and correlation names.

You can save the query as a view, or you can create a new table from the columns.

In R:BASE for Windows QUERY starts the **Query By Example** option from the **Tools** menu. After you enter the command QUERY, what R:BASE displays depends on the options you specified.

To display...	Do the following...
The "Query By Example" window	Use the QUERY command without specifying any options.
The "Query By Example" window and the query contained in the QUERY command	Use the QUERY command with any option except EXECUTE.
The Data Browser with the data retrieved by the QUERY command	Use the QUERY command with EXECUTE and any other option.

Examples

The following command chooses all the columns from the *customertable* and displays the columns' names and data on the screen.


```
QUERY * FROM customer EXECUTE
```

The next command selects the *empid*, *empfname*, and *emplname* columns from the *employee* table, and displays the column names on the screen.

```
QUERY empid, empfname, emplname FROM employee
```

The following command selects the data from the *custid* column in the *customer* table and the matching *netamount* columns in the *transmaster* table. *T1* and *t2* are correlation names. In the WHERE clause, the link is established between the two tables by comparing the values of the *custid* column between the two tables. The linking columns can have different names; however, they must have the same data type.

```
QUERY t1.custid, t2.netamount FROM customer t1, transmaster t2 +  
WHERE t1.custid = t2.custid EXECUTE
```

1.18.3 QUIT

In a program, use QUIT to close all open program files, [IF/ENDIF](#) structures, [SWITCH/ENDSWITCH](#) structures, and [WHILE/ENDWHILE](#) loops.

```
QUIT [ TO filespec ]
```

Option

TO filespec

Specifies the command file that is to run after the QUIT command executes. QUIT TO another code-locked binary command file is also supported. QUIT is only supported with external files and cannot be used with Custom EEPs.

About the QUIT Command

When the QUIT command runs, it clears the nesting levels R:BASE maintains to control the following two items:

- Nesting command files called by the [RUN](#) commands.
- Levels for nesting [IF](#) structures, [SWITCH](#) structures, and [WHILE](#) loops.

Note:

- The QUIT command is not supported in R:Compiler for R:BASE.

Examples

The following command closes all open command files, IF structures, SWITCH structures, and WHILE loops, then you are returned to the R> Prompt.

```
QUIT
```

The command below closes all open command files, IF structures, SWITCH structures, and WHILE loops, and runs the command file named GETVAL.INV.

```
QUIT TO getval.inv
```

1.19 R

1.19.1 RBAPP

Use the RBAPP command to start the Application Designer with which you can create or edit RBA application files.

```
RBAPP [ filespec ]
```

filespec

Specifies the R:BASE application file to be opened in the Application Designer.

Example:

```
RBAPP TestApp.rba
```

1.19.2 RBBEDIT

Use the RBBEDIT command to start the R:BASE BLOB Editor, which can be used to create and edit external files at the R> Prompt, in a command file, or in an EEP.

```
RBBEDIT [ filename.ext [ ZoomMode [ DefaultTab ] [ MDI ] ] ]
        [ '<SQL query>' ]
```

Options

DefaultTab

Specifies the default tab for the R:BASE BLOB Editor.

Default Tab	Parameter
Image	IMAGE
Text	NOTE, VARCHAR, TEXT, NOTEVARCHAR
Rich Text	RICHTEXT, RTF
Unicode Text	UNICODEBSTR, UNICODE, BSTR
Hex	HEX

filename.ext

Specifies an external file.

MDI

Specifies a modeless window, which allows you to access other windows without closing the BLOB Editor window first.

<'SQL query'>

Specifies the SELECT command to edit a BLOB directly from a table. The command mimics the behavior of double clicking of a BLOB field in the Data Browser.

ZoomMode

Specifies the zoom mode for the displayed image file

Zoom Parameter	Description
ACTUAL	zooms the image to the actual size
FULL	fits the image to the full page of the BLOB Editor window
FIT_WIDTH	fits the image to the current BLOB Editor window width

FIT_HEIGHT	fits the image to the current BLOB Editor window height
FIT_PAGE	fits the image to the current BLOB Editor window

Examples:

01.

The following will display the BLOB Editor.

```
RBBEDIT
```

02.

The following will display the image in the "Images" tab of the BLOB Editor in the full page zoom mode.

```
RBBEDIT myphoto.bmp FULL IMAGE
```

03.

The following will display the "Rich Text" tab of the BLOB Editor on a modeless window. If the file exists it will be loaded. The RICHTEXT parameter forces the "Rich Text" tab to be displayed.

```
RBBEDIT myfile.rtf RICHTEXT MDI
```

04.

The following will display the text file in the "Text" tab of the BLOB Editor.

```
RBBEDIT mytext.txt TEXT
```

05.

The following will display the file in the "Unicode Text" tab of the BLOB Editor.

```
RBBEDIT filename UNICODE
```

06.

The following will display the contact photo image in the BLOB Editor.

```
RBBEDIT '<SQL SELECT ContactPhoto FROM Contact WHERE ContID = 1001>'
```

1.19.3 RBDEFINE

Use the RBDEFINE command to start the Data Designer.

RBDEFINE

About the RBDEFINE Command

Use the Data Designer to create or modify an existing table and its constraints, rules, indexes, and keys.

Alternative R:BASE Commands

Use the following commands as an alternative to the Data Designer:

- [CREATE SCHEMA](#)
- [CREATE TABLE](#)

Use the following R:BASE commands to enhance or modify a database:

- [ALTER TABLE](#)
- [COMMENT ON](#)
- [CREATE INDEX](#)
- [CREATE VIEW](#)
- [DELETE](#)
- [DROP](#)

- [LIST](#)
- [RENAME](#)
- [RULES](#)

1.19.4 RBEDIT

Use the RBEDIT command to start the R:BASE Editor with which you can create or edit text files and command files.

```
RBEDIT [ filespec [ SELECT VARCHAR clause ] ]
```

Options

filespec

Displays the contents of the file on the screen for editing. If the file does not exist, a blank screen is displayed. If you specify a wildcard with the *filespec* option, RBEDIT displays a list of all files that meet the condition. You can then choose a file from the list.

SELECT VARCHAR clause

Specifies a column defined with the VARCHAR data type from a table, then you can edit the contents. The [SELECT](#) clause must limit the data to only one row; otherwise, an error is returned.

About the RBEDIT Command

You can start the R:BASE editor by entering RBEDIT at the R> Prompt, or by choosing **R:BASE Editor** from the **Tools** menu in R:BASE.

Example

The following command starts REdit and displays the file named OLDFILE for editing. If OLDFILE does not exist, the editor creates a new file and names it OLDFILE.

```
RBEDIT oldfile
```

To launch the help from within the R:BASE Editor, press [Shift]+[F1].

1.19.5 RBLABELS

Use the RBLABELS command to launch the Label Designer to create and edit labels.

```
RBLABELS [ lblname [ MODAL tblview ] ]
```

Options

labelname

Specifies the name of an existing label to modify.

MODAL

Launches a label in the designer from a command block, command file, or EEP. The Label Designer must be closed in order to continue the next command.

tblview

This option is for DOS only; names the first table or view that a label serves. Use this option when you are creating a new label. This option is available only if you start to build a label from the R> Prompt with a database open. Specify both *labelname* and *tblview* when you want to build a new label. R:BASE displays the Label Designer screen, where you can begin to define the new label. Use *labelname* alone to specify the name of an existing label you want to modify.

About the RBLABELS Command

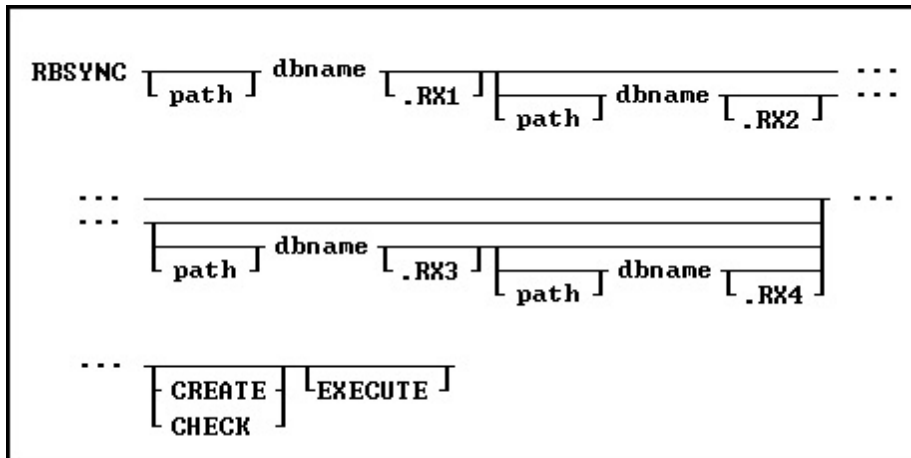
You can display the Label Designer with the RBLABELS command.

When creating labels, you can do the following:

- Build a label definition to create labels from data in a table or view.
- Set dimensions of a label definition to fit all popular label stock.
- Define variables and include them on your labels.
- Print labels on a printer, display them on screen, or send them to a file.

1.19.6 RBSYNC

Use the RBSYNC command to synchronize the timestamps for database files.



Options

dbname

Specifies the name of the database to synch.

path

Specifies the location of the database file(s).

.RX1

Specifies the .RX1 database file.

.RX2

Specifies the .RX2 database file.

.RX3

Specifies the .RX3 database file.

.RX4

Specifies the .RX4 database file.

CREATE

Creates a skeleton index file if it can't be found.

CHECK

Shows current synch status only. This is the default option unless you specify otherwise.

EXECUTE

Executes the command to synch the database.

About the RBSYNC Command

Use the RBSYNC command to synchronize the date and timestamps for the database files. The database files can become out of synch when R:BASE is not exited normally. You cannot open a database if the database files are out of sync.

With RBSYNC, a database owner can update the encoded timestamp values in the four database files to all have the same timestamp values. This might be necessary if R:BASE was aborted or if the computer or operating system crashes.

When a database is created or opened, a timestamp is encoded in each of the four files. Encoding the timestamp in each of the four database files enables the database files to be stored in separate directories apart from one another. To ensure that the correct database files are accessed, R:BASE evaluates the timestamp values of the files as it searches for them on the path statement. R:BASE must first try to connect to File 1, and then it searches the path for the other database files.

When you start RBSYNC, you can choose the location of File 1, enter the database owner's user identifier if one exists, then RBSYNC displays the location and timestamp for all four database files.

RBSYNC displays error messages if the database is already open or if any of the three files can't be located; however, if File 1 and File 2 are located but not File 3, the user has the option to create a new (empty) File 3 or abort. If creating a new File 3, RBSYNC displays a warning message stating that a PACK operation should be performed.

Notes:

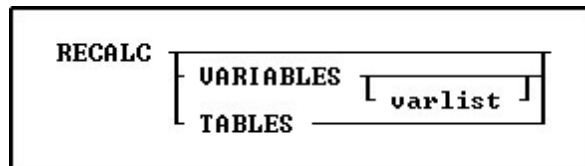
- RBSYNC requires the EXECUTE parameter to actually perform the synchronization.
- You can implement the [AUTOSYNC](#) operating condition automatically synchronize the database files, if necessary.
- The *path*, *dbname*, and *.RX** options are rarely used except when the database files are in different locations.

Example:

```
RBSYNC dbname EXECUTE
```

1.19.7 RECALC

Use the RECALC command in an entry/exit procedure (EEP) to recalculate expressions in a form.



Options

TABLES

Finds, recalculates, and redraws the lower-level tables to reflect the actions of the entry/exit procedure (EEP). The recalculations will be performed only for the active table (the table of the currently focused control).

VARIABLES

Recalculates all variables for the current table.

VARIABLES varlist

Recalculates a list of one or more variables. Use this option at the end of a set of procedures to recalculate variables.

About the RECALC Command

Without using the RECALC Command in EEPs-A recalculation takes place when you move to the field following the field that triggered the EEP. R:BASE automatically redisplay the value in a form field (variable) that calls the EEP. Redisplaying the value in a form field ensures that the field always contains the correct value.

To automatically redisplay a field, the following statements must be true:

- The EEP must be started from the field you want redisplayed.
- The field you want redisplayed must be a variable.
- The EEP loads a new value into a variable that is the same as the variable that started the EEP.

If you want to load the value into a column, create a form expression that equates the variable and the column, for example, *colname=.varname*.

Using the RECALC Command in EEPs-Use RECALC to ensure that expressions based on variables modified by an EEP are always current. RECALC only updates expressions when the following conditions apply:

- The field containing the EEP is a variable.
- The EEP changes the value of the variable.
- The EEP runs the RECALC command before returning to the form.
- At least one form expression has the variable to the right of the equal sign, which signifies a dependent expression.

When you use the RECALC command without any options, RECALC redisplay only the variable in the field that started the EEP, and recalculates only expressions using that variable. Use the VARIABLES option to force redisplay of all variables placed on the form and recalculation of all table expressions. Use the TABLES option to force recalculation of expressions in the other tables used in a multi-table form. The TABLES option recalculates lower tables only, not the current table.

Examples

Example 01:

If you have an EEP in a form that calculates a value for a variable that you want to load into a column, follow the procedure below:

To calculate a value for a variable to load into a column:

1. Place the variable on the form.
2. Include the EEP in the field settings for the variable.
3. Create an expression that equates the column to the variable so that the variable is loaded into the column.
For example: *colname1 = (.varname1)*
4. Include RECALC or RECALC VARIABLES in the EEP.

For example, a procedure that calculates the sum from a column could be:

```
SELECT SUM colname2 INTO varname1 FROM tblname WHERE +
colname3 = .varname2
RECALC VARIABLES
RETURN
```

Example 02:

--Recalculates the values for the variables *vTargetPrice*, *vShippedQuantity*, and *vDiscountedRate*
 RECALC VARIABLES *vTargetPrice*, *vShippedQuantity*, *vDiscountedRate*

Example 03:

--Recalculates data for lower-level tables
 RECALC TABLES

1.19.8 RECOVER

Use the RECOVER command to start the Recover utility, which corrects errors that occur during transaction processing, for instance from a network or power failure.

```
RECOVER [dbname]
```

Options

dbname

Specifies the database to recover.

About the RECOVER Command

You can use the RECOVER command to restabilize a database after a transaction is interrupted. The following conditions can interrupt a transaction:

- R:BASE discovers minor inconsistencies or unresolvable resource conflicts within a transaction. R:BASE automatically rolls back the transaction and displays an error message.
- Transaction processing has been interrupted unexpectedly. When you try to connect to the database, R:BASE detects either an existing Before Image file (a log of the commands issued in a transaction) or major inconsistencies. An error message tells you to use the RECOVER command.

Other users connected to the same database might get the same message, but not until they disconnect from the database and try to reconnect to it. The transactions entered by other users still connected to the database are executed properly unless they try to access system resources (such as tables and views) that are directly affected by the transaction error.

When transaction processing is on, R:BASE creates Before Image files for the current database. When you use the RECOVER command, it rolls back the interrupted transaction by using each Before Image file, clears all table and database locks, and resolves any other internal inconsistencies.

Before you use the RECOVER command, do the following:

- Backup your database and Before Image files. The Before Image files are in the same directory as the database and have extensions that begin with "B," such as DATABASE.B01.
- Make sure all users have exited from the affected database.

1.19.9 REFF

Use the REFF command to run External Form Files.

```
REFF [filename.rff] [MDI [AS alias]] [CAPTION 'windowcaption']
```

filename.rff

Specifies the name of the external form file to launch.

MDI

Specifies a modeless form, allowing users to access other windows without closing the form first. This option is used in R:BASE for Windows only.

AS alias

This option is used in R:BASE for Windows only. Allows the user to specify a name for this instance of the form. This option is used in conjunction with the MDI option so that the user can specify the form with the [SETFOCUS](#) command.

CAPTION 'windowcaption'

This option is used in R:BASE for Windows only. Specifies a caption for the window title bar.

Examples

```
REFF RRBYW20.RFF CAPTION 'Running R:BASE Your Way!'
```

```
REFF AdhocReports.RFF MDI AS Adhoc CAPTION 'Adhoc Custom Reports'
```

1.19.10 RELOAD

Use the RELOAD command to copy an open database without copying any unusable space.

RELOAD dbspec [WITH USER CASE]

Options**dbspec**

Specifies the new database name.

WITH USER CASE

Incorporates case folding and collating tables defined in the user's configuration file into the reloaded database.

About the RELOAD Command

Disk space becomes unavailable in a database when the following actions are performed:

- Deleting rows or indexes.
- Removing tables or columns.
- Adding columns or modifying tables with the [ALTER TABLE](#) command.

RELOAD copies a database table by table, and places all rows for each table in a single area on the disk, which improves database-response time.

When you use RELOAD to reload a database on the same disk and directory as the original database, enter a different name for the new database. When you reload a database from a different disk or directory onto the current disk, you can use the same database name for the copy. Be sure to specify the new drive or directory when you enter the command.

If there is not enough available disk space to copy a database using the RELOAD command, use the [PACK](#) command instead. Backup the database before packing. PACK eliminates unused space in a database; however, PACK does not rearrange the rows-only the RELOAD command rearranges rows.

RELOAD is available when [MULTI](#) is set on and a user has not set any locks on the database.

RELOAD is unavailable when Transaction Processing is on.

When you reload data that has a NOTE data type, the rows are adjusted according to the current setting of the [SET NOTE PAD](#) command.

RELOAD is unavailable when [STATICDB](#) is set on, which activates a read-only schema mode.

Database Access Rights with RELOAD

When access rights for a table have been assigned using the [GRANT](#) command, RELOAD requires the database owner's user identifier to RELOAD a database.

Example

The following command reloads an open database and gives it the name *newbase* in the RBASE directory of drive C:.

```
RELOAD c:\rbase/newbase
```

1.19.11 RENAME

Use the RENAME command to change a form, report, label, table, view, or column name, and the database owner's user identifier. You can also use RENAME to change the name of an existing file.

```

      COLUMN colname1 TO colname2 [ IN tblname ] [ NOCHECK ]
      TABLE tblname1 TO tblname2
      VIEW viewname1 TO viewname2
RENAME FORM formname1 TO formname2
      LABEL labelname1 TO labelname2
      PROCEDURE procname1 TO procname2
      REPORT rptname1 TO rptname2
      OWNER ownername1 TO ownername2

RENAME filespec filename

```

Options

COLUMN colname1 TO colname2

Renames a column in one table or in all tables in the open database.

filename

Specifies the new name of the file. Wildcards can be used.

filespec

Specifies the file you want to rename. Optionally, include a drive and path specification in the form D:\PATHNAME\FILENAME.EXT. Wildcards can be used.

FORM formname1 TO formname2

Renames a form in the open database.

IN tblname

Specifies the table in which you want to rename a column.

LABEL labelname1 TO labelname2

Renames a label in the open database.

NOCHECK

Does not update references to views, tables, and columns in forms, reports, labels, access rights, and rules.

OWNER ownername1 TO ownername2

Renames an owner in the open database.

PROCEDURE procname1 TO procname2

Renames a stored procedure in the open database.

REPORT rptname1 TO rptname2

Renames a report in the open database.

TABLE tblname1 TO tblname2

Renames a table in the open database.

VIEW viewname1 TO viewname2

Renames a view in the open database.

About the RENAME Command

If you do not want R:BASE to update references to views, tables, and columns when you rename them, include the NOCHECK option with the command.

Renaming Columns

You can rename a column in an entire database or in a single table. R:BASE does not update column references in rules. When you rename a column, R:BASE automatically updates references to the column in the following instances:

- If the column has a description.
- If it is used in a form, report, label, computed column, [UPDATE access right](#), or autonumbered column. However, because of possible size problems, R:BASE does not change column references inside an expression in a form, report, or label. These column references must be modified manually through the Form, Report, or Label Designer.

Updating Views and Tables

R:BASE automatically updates references to views and tables in the following instances:

- If you rename a view used in a report, label, or access right.
- If you rename a table that has a description, or is used in a form, report, label, or access right.

R:BASE does not update table or column references in views.

To update a view, delete it with the [DROP](#) command and define it again with the [CREATE VIEW](#) command, or QBE. To update a rule, you can use the [RULES](#) command, or the **Database Designer**. If you use the RULES command, you must first delete the rule with the DROP command and then add it again with the RULES command.

Renaming Tables

When you rename a table that is used as the rule table in a data-entry rule, R:BASE updates the rule definition. However, if you rename a table used in the WHERE clause of a rule definition, you must update the rule yourself. R:BASE does not update table references in views.

Renaming Files

When you use RENAME to change the name of a disk file, only the name of the file is changed. The file remains in the same directory on the same drive. You can include a file specification for the file you are renaming but not for the new file name. If you want the file to reside in a different drive or directory, use the [COPY](#) command. This command is similar to the operating system command RENAME.

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

Updating Command Files

R:BASE also does not update column, table, view, form, report, or label references in command files or applications. To update command files, use REdit or another text editor. To update applications, use the Application Designer.

Assigning User Identifiers

You can assign or change the database owner's user identifier with RENAME. The default user identifier is PUBLIC. Until this default is changed, any user can modify the database structure, read, enter, change, or delete data.

An owner's user identifier can be a maximum of 36 characters. It must begin with a letter, and can contain letters, numbers, and the symbols #, \$, _, and %, and must be unique among all user identifiers.

Examples

The following command renames a column from *transid* to *transxno* in the *transmaster* table.

```
RENAME COLUMN transid TO transxno IN transmaster
```

The following command changes the database owner's user identifier from the default PUBLIC to *june*.

```
RENAME OWNER PUBLIC TO june
```

The following command changes the name of the CUSTOMER file to CUSTOMER.DAT.

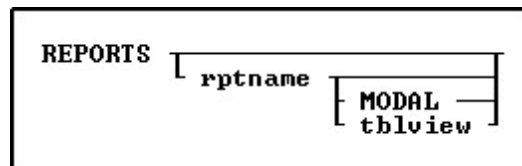
```
RENAME customer customer.dat
```

The following command renames all four database files to NYC.RX1, NYC.RX2, NYC.RX3, and NYC.RX4.

```
RENAME newyork.rx? nyc.rx?
```

1.19.12 REPORTS

Use the REPORTS command to start the Report Designer.



Options

MODAL

Launches a report in the designer from a command block, command file, or EEP. The Report Designer must be closed in order to continue the next command.

rptname

Specifies an existing report whose definition you want to modify.

tblview

This options is for DOS only. Specify both *rptname* and *tblview* when you want to build a new report. R:BASE displays the Report Designer window where you can begin to define the new report. Use *rptname* alone to specify the name of an existing report you want to modify.

About the REPORTS Command

When designing a report, the following can be performed:

- Define a Quick or a Custom report that prints or displays data from a table or view
- Group data by unique column values with breaks
- Define variables and include them in the report
- Look up values from columns in other tables
- Draw lines, borders, 3D boxes, and check boxes
- Print a report to a printer, the screen, or a file
- Print summary information, such as subtotals
- Create display formats for data
- Place a graphic image

- Create a sub report
- Display barcodes and 2-dimensional barcodes
- Create a PDF form report
- Place a page style background for all pages

Example

The following command launches the *Invoice* report within the Report Designer.

```
REPORTS Invoice
```

1.19.13 RESET

The RESET command adjusts the last modification date/time stamp for tables and views.

```
RESET tblview LAST_MOD TO value
```

When copying table and view information with the [UNLOAD](#) command, the output automatically generates RESET commands after the table/view definition, to reset the date/time stamp and retain the original value.

tblview

Specifies a table or view.

value

Specifies the date/time stamp for tables or views.

Examples

The below resets the last modification date and time stamp for the *LicenseInformation* table.

```
RESET LicenseInformation LAST_MOD TO '12/15/2022 12:00'
```

The below resets the last modification date and time stamp for the *SalesDataByCompany* view.

```
RESET SalesDataByCompany LAST_MOD TO '01/01/2022 08:00'
```

1.19.14 RESETROW

Use the RESETROW command to reset the current row of data currently being edited or entered in a form. This will cause the data to revert to it's original format and content.

```
RESETROW
```

1.19.15 RESTORE

Use the RESTORE command to restore part or all of a database from a file created with [BACKUP](#).

```
RESTORE filespec
```

Option

filespec

Specifies a file name with a drive and path specification in the form D:\PATHNAME/FILENAME.EXT.

About the RESTORE Command

If you are restoring a backup copy from a floppy disk, insert the disk in the floppy drive. If there is more than one backup disk, insert the disk labeled 1 first. RESTORE prompts you for the appropriate disk and allows you to confirm each disk. If you backed up data and structure separately, you can choose not to restore any subsequent disks. RESTORE appends data to the table if it exists in the database.

If your database becomes damaged after your last backup and any of the .RBN files of the database remain, you can rename the database files, restore the database on a different directory, or erase them before you restore the database from the backup.

When [STATICDB](#) is set on, which activates a read-only schema mode, RESTORE is unavailable.

To use RESTORE, the RESTORE.RMD file must be in the directory containing the R:BASE program files. If you cannot locate the RESTORE.RMD file in your R:BASE program directory, then save the following as the file.

```
--RESTORE.RMD
CLEAR VARIABLES YesNo, OldQuote, Quote
SET VARIABLE Quote TEXT = QUOTES
SET VARIABLE OldQuote = (CVAL(.Quote))
SET QUOTES=
SET VARIABLE YesNo = 'Y'
WHILE YesNo <> 'E' THEN
    BEEP
    SET QUOTES=
    WRITE 'Please insert disk into drive...'
    DIALOG 'Press Enter to continue, or S to stop operation.' YesNo vEndKey 1
    IF YesNo = 'S' THEN
        SET QUOTES = .OldQuote
        PAUSE 1
        RETURN
    ENDIF
    INPUT .%1
ENDWHILE
PAUSE 1
RETURN
```

Examples

This command restores the *concomp* database from the CONCOMP.BUP file on drive A:.

```
RESTORE a:concomp.bup
```

The following command lines open the database and restore the TBLNAME.BUP file on drive A: that contains a backup of a table. R:BASE generated the TBLNAME.BUP file when you used the [BACKUP](#) DATA command.

```
CONNECT dbname
RESTORE a:table.bup
```

1.19.16 RETURN

Use the RETURN command in a program to return control to a program or to the R> Prompt.

RETURN `[.varname]`

Options

.varname

Used ONLY within a stored procedure to return a value. The returned value will be stored in the STP_RETURN system variable. This option will return an error when used outside of a stored procedure. The default is text 8 characters, but if you want more, set it to a larger value.

[PUT](#) Command Example (sets the limit for the return value at 30 characters):

```
PUT test.prc AS TEXT test p1 integer RETURN TEXT (30)
```

About the RETURN Command

When you use RETURN to return to a command file or a block in a procedure file, the next command executed is the command following [RUN](#) in the command file or procedure file.

If you used the RUN commands to run a command file or a block in a procedure file from the R> Prompt, control returns to the R> Prompt when RETURN is executed.

Examples

Assume you have two command files named CMDFIL1.RMD and CMDFIL2.RMD with the commands shown below.

```
--CMDFIL1.RMD
PAUSE 2 USING 'This command file (cmdfil1.rmd) calls cmdfil2.rmd.'
RUN cmdfil2.rmd
CLS
PAUSE 2 USING 'Cmdfil1.rmd is in control.'
```

```
--CMDFIL2.RMD
LABEL top
DIALOG 'Press Q to quit:' vend vkey 1
IF vend = 'Q' THEN
    RETURN
ELSE
    GOTO top
ENDIF
```

When you run CMDFIL1.RMD by entering RUN CMDFIL1.RMD, the following messages appear on the screen:

```
This command file (cmdfil1.rmd) calls cmdfil2.rmd.
```

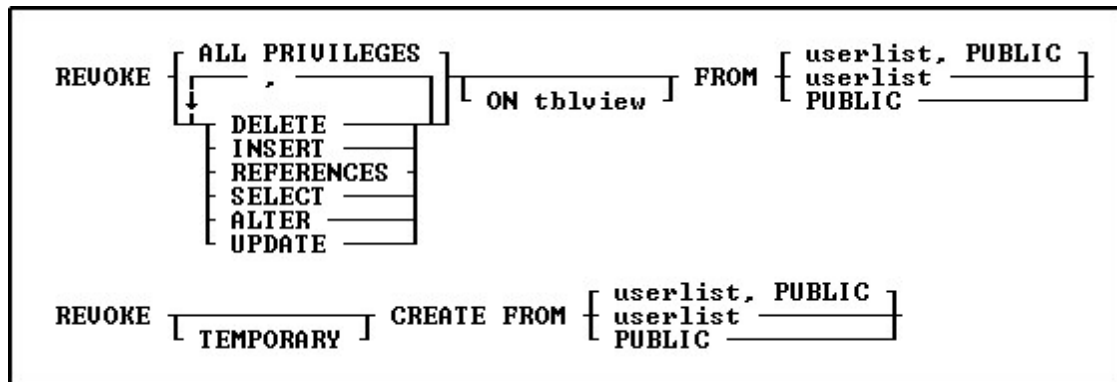
```
Press Q to quit:
```

If you press [Q] at the [DIALOG](#) command prompt, control returns to CMDFIL1.RMD. The following message appears on screen, processing stops, and you are returned to the R> Prompt:

```
Cmdfil1.rmd is in control.
```

1.19.17 REVOKE

Use the REVOKE command to remove privileges provided to users with the [GRANT](#) command.



Options

' Indicates that this part of the command is repeatable.

ALL PRIVILEGES

Removes all user privileges granted for all tables and views or for one table or view.

ALTER

Removes permission from users to modify the structure of all tables or specified tables.

CREATE

Removes permission from users to create new tables. Do not specify any tables or views when removing this permission.

DELETE

Removes permission to remove rows from all tables and views, or from a specified table or view.

FROM PUBLIC

Specifies PUBLIC. If, for example, *Ralph*, *Sam*, *Jane*, and PUBLIC have been granted certain user privileges, revoking privileges from PUBLIC would not affect the three listed users.

FROM userlist

Specifies individual users whose access is to be revoked. You must separate user identifiers with a comma (or the current delimiter). For a value with spaces, the userid must be enclosed in quotes.

FROM userlist, PUBLIC

Specifies both individual users and PUBLIC, whose access is to be revoked. You must separate user identifiers with a comma (or the current delimiter). For a value with spaces, the userid must be enclosed in quotes.

INSERT

Removes permission to add rows to all tables and views or to a specified table or view.

ON tblview

Specifies a table or view from which to remove user privileges.

REFERENCES

Removes permission to create a table with a foreign key that references a table with a primary key.

SELECT

Removes permission to view and print data from all tables and views, or from a specified table or view.

TEMPORARY

Removes permission from users to create new temporary tables.

UPDATE

Removes permission to change the value of all columns on all tables and views, or on a specified table or view. You cannot specify columns when revoking UPDATE permission.

About the REVOKE Command

If you are the owner of a database, you can revoke any user privileges granted to users. If the database owner or other users have assigned you user privileges with the [WITH GRANT OPTION](#), you can revoke only the user privileges that you have granted to other users.

To remove the WITH GRANT OPTION, you must first revoke the privilege(s) to which the WITH GRANT OPTION has been assigned. Use the [LIST ACCESS](#) command to display a list of user privileges. An asterisk before an user privilege indicates that the WITH GRANT OPTION has been assigned to that user privilege, for example, *UPDATE.

REVOKE ALL PRIVILEGES revokes all user privileges that have been granted. However, REVOKE combined with ALTER, CREATE, DELETE, INSERT, REFERENCES, SELECT or UPDATE only applies to those privileges.

You can remove more than one user privilege in a REVOKE command. Separate the user privileges with a comma (or the current delimiter). You can also revoke user privileges using the **User Privileges** option from the **Utilities** menu.

Examples

Assume that the following sequence of [GRANT](#) commands represents all the user privileges granted for the *ConComp* database.

```
GRANT INSERT ON Employee TO Ralph, Sam
GRANT SELECT, INSERT ON TransMaster TO Jane WITH GRANT OPTION
GRANT UPDATE ON TransMaster TO Sam
GRANT UPDATE (Company, CustAddress, CustCity) ON Customer TO Sam, PUBLIC
```

The following command revokes permission granted to *Jane* to display or print data, or add rows to the *TransMaster* table.

```
REVOKE SELECT, INSERT ON TransMaster FROM Jane
```

The following command revokes the UPDATE user privilege granted to *Sam* for all tables and views in the database.

```
REVOKE UPDATE FROM Sam
```

The following command revokes all user privileges granted to *Sam*, except those granted to him as a member of PUBLIC.

```
REVOKE ALL PRIVILEGES FROM Sam
```

The following command revokes all user privileges for all tables and views for the users *Sam*, *Jane*, and *Ralph*; and the PUBLIC account.

```
REVOKE ALL PRIVILEGES FROM Sam, Jane, Ralph, PUBLIC
```

The following command revokes CREATE privileges for temporary tables for the user.

```
REVOKE TEMPORARY CREATE FROM Noah
```

1.19.18 RHELP

Included with R:BASE 2000 (version 6.5+++) is the external file RHelp.EXE. This file can be called using the [ZIP](#) command and will display the contents of the file specified. Users will not be able to edit the file displayed. This window can be Minimized, Normalized and Maximized with an option to Close [x]. **The RHelp.EXE is no longer provided with R:BASE.**

ZIP RHELP filespec

Options

filespec

Displays the contents of the file on the screen. This may be either a plain text file or a Rich Text Format (RTF) file. If the file name includes spaces you must enclose the *filespec* in either single or double quotes.

Example

The following command starts RHelp and displays the file named RichFile.RTF for viewing.

```
ZIP RHELP C:\RBTI\RichFile.RTF
```

The following example displays the Long File Named file "This is a Test.RTF" from a network share "Users" on the server "InHouse".

```
ZIP RHELP "\\InHouse\Users\This Is a Test.RTF"
```

1.19.19 RHIDE

Use RHIDE to hide the background R> Prompt window when using modal forms.

RHIDE

The RHIDE command is specific to EEPs.

1.19.20 RMDIR

Use the RMDIR command to remove a directory from a drive.

RMDIR [d:] pathname

Options

d:

Specifies a drive letter.

pathname

Specifies one or more directory names that lead from the root directory to a specified directory.

About the RMDIR Command

The R:BASE RMDIR command is similar to the operating system RMDIR command. For more information about the RMDIR command, refer to your operating system manual.

You cannot remove the root directory or the current directory. Before you can remove a directory, all files and directories in that directory must be removed. To delete files from a directory, use the [ERASE](#) command. When a file path includes more than one directory, the last directory listed is removed.

Examples

The following command removes the OLDDATA directory from the current drive.

```
RMDIR olddata
```

The following command removes the OLDDATA directory from the /SALESDEP/YTDSALES file path on drive C:.

```
RD c:\salesdep\ytdsales/olddata
```

1.19.21 ROLLBACK

Use the ROLLBACK command to cancel the changes made to the data or database structure made since the beginning of the transaction, when using transaction processing.

ROLLBACK [WORK]

Options

WORK

WORK is optional and has no effect in R:BASE.

About the ROLLBACK Command

A transaction consists of all the commands executed since the last [COMMIT](#) or ROLLBACK command, or since you connected to the database if a COMMIT command has not been executed.

If you want to be able to process a group of commands, or transaction, without permanently affecting your data until you accept the changes, turn the [AUTOCOMMIT](#) setting off. R:BASE stores the information needed to reverse each modification made to the database in a Before Image file.

To reverse a transaction and undo the changes, enter the ROLLBACK command after you have entered the transaction. The ROLLBACK command also releases all automatic table and database locks. If you want to make the changes permanent, enter the COMMIT command.

This command is only valid when you open a database with [TRANSACTION](#) set on.

1.19.22 RPHONE

The RPHONE command uses the Windows Phone Dialer to dial a call. You must have a modem installed in your computer.

RPHONE 'phonenum' ['nameforwindowslog' ['comment']]
--

Options

nameforwindowslog

Optionally pass in the name of the party you are calling to the Windows Dialer call log.

comment

Optionally pass a comment about the phone call into the Windows Dialer call log.

About the RPHONE Command

RPHONE command uses the Windows Phone Dialer to dial a call. You must have a modem installed in your computer.

RPHONE command requires the number that you want to call. You may optionally pass in the name of the party you are calling and a comment about the call. This information will be placed in the Windows Dialer call log if the call is successful.

The RPHONE command is also supported in EEPs.

Examples

Example 01:

```
RPHONE '(800) 555-2368'
```

Example 02:

```
RPHONE '(800) 555-2368' 'Ghostbusters' 'Call Now'
```

Example 03:

```
SELECT CustPhone, Company INTO +
vCustPhone INDIC IvCustPhone, +
vCompany INDIC IvCompany +
FROM Customer WHERE CustID = .vCustID
RPHONE .vCustPhone .vCompany 'Follow-Up'
```

1.19.23 RSHOW

Use RSHOW to bring back the background R> Prompt window when using modal forms.

RSHOW

The RSHOW command is specific to EEPs.

1.19.24 RULES

Use the RULES command to regulate data entry in a database.

RULES 'message' FOR tblname {
SUCCEEDS
DELETE
 } [
FAILS
WHERE clause
]

Options

DELETE

- **DELETE SUCCEEDS** deletes a row from a database when the conditions in the [WHERE](#) clause are met.
- **DELETE FAILS** deletes a row from a database when the conditions in the WHERE clause are not met.

FOR tblname

Specifies the name of the table for which you are defining rules.

FAILS

Specifies that a row must not meet the conditions included in the WHERE clause in order to be added to the database.

'message'

Specifies a message to be displayed when a rule is violated.

SUCCEEDS

Specifies that a row must meet the conditions included in the WHERE clause in order to be added to the database.

WHERE clause

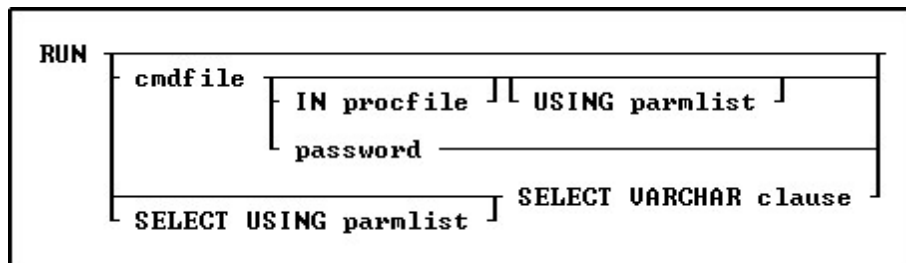
Limits rows of data. For more information, see the WHERE command.

Example:

```
RULES 'Component Must Exist in Components table.' +
FOR CompUsed SUCCEEDS +
WHERE EXISTS +
(SELECT CompID FROM Component t2 +
WHERE t2.CompID = CompUsed.CompID)
```

1.19.25 RUN

Use the RUN command to run command blocks, command files, and command files requiring passed parameters.

**Options****cmdfile**

Specifies the name of the command block or command file to execute.

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

IN procfile

Specifies the name of a procedure file. A procedure file is a compiled binary file that contains stored menu, screen, and command blocks. Include a drive and path name if the procedure file is not on the current drive and directory.

password

Specifies the file password if encryption was specified when the file was saved or created. See [OUTPUT](#) for the command syntax to encrypt database information output. See R:BASE Editor for information on saving encrypted command files.

SELECT VARCHAR clause

Specifies a column defined with the VARCHAR data type from a table, then you can run the contents. The SELECT clause must limit the data to only one row; otherwise, an error is returned.

USING parmlist

Lists the values the command file uses when it runs. The parameter list can contain up to 18 values. The first value in the list is referenced in the executed file as %1, the second as %2, and so on through %9. They are treated just like other variables. To reference the contents of these variables, preface the variable name with a dot (.); for example, SET VAR v1 =.%1.

About the RUN Command

The RUN command must be on a line by itself and not combined with other commands.

Examples

The following command runs a file named MYCMD.CMD in the current working directory.

```
RUN mycmd.cmd
```

The following command runs a command block named *mycmdfil* in the MYPROCFL.APX procedure file.

```
RUN mycmdfil IN myprocfl.apx
```

The following command executes the *mycmdfil* command block in the MYPROCFL.APX procedure file, placing the parameter values, *Display This Message* and *10*, into parameter variables %1 and %2, respectively.

```
RUN mycmdfil.rmd USING 'Display This Message' 10
```

The following commands, which use the system variable *#time*, make up a timing procedure called *mycmdfil.rmd*. This procedure displays the message passed as parameter %1 for the length of time indicated in parameter %2. When executing the RUN command at the R> Prompt, it causes *mycmdfil.rmd* to display the message for 10 seconds.

```
--mycmdfil.rmd
SET VARIABLE vstart TIME = .#TIME
SET VARIABLE vwait INTEGER = 0
WHILE vwait < .%2 THEN
    SHOW VARIABLE %1
    SET VARIABLE vwait = (.#TIME - .vstart)
ENDWHILE
CLS
CLEAR VARIABLES vstart, vwait, %1, %2
RETURN
```

With applications that frequently pass parameters with the RUN command, it is recommended to set variables to the value of the passed parameters within the command file being RUN, and then clear the passed parameters. This practice will help avoid data type conflicts and assists in clearing parameter variables.

The [CLEAR VAR](#) command below will clear all existing % variables (assuming your [SINGLE](#) character setting is an underscore). The CLEAR VAR uses 2 underscores, a dash, and an underscore. The passed parameter variables will display in [TRACE](#) mode as %1-1 in a single nesting level. If a called command file calls another, the variable would then be %1-2. The x in "%1-x" is the variable nesting level.

```
RUN cmdfile.rmd USING 'Display This Message' 10
```

```
--cmdfile.rmd
SET VARIABLE vstart TIME = .#TIME
SET VARIABLE vwait INTEGER = 0
SET VARIABLE vmessage TEXT = .%1
SET VARIABLE vmax INTEGER = .%2
CLEAR VAR __-__
SET VARIABLE
WHILE vwait < .vmax THEN
```

```

SHOW VARIABLE .vmessage
SET VARIABLE vwait = (.#TIME - .vstart)
ENDWHILE
CLS
CLEAR VARIABLES vstart, vwait, %1, %2
RETURN

```

The following RUN command executes the command syntax with the *CmdData* VARCHAR data field inside the table *IntrnlCmd*.

```
RUN SELECT CmdData FROM IntrnlCmd WHERE CmdName = 'NewKey'
```

1.19.26 RUNRBA

Use the RUNRBA command to run .RBA application files, which are created specifically in the Application Designer.

RUNRBA filespec

filespec

Specifies the R:BASE application file to be run.

Example:

```
RUNRBA TestApp.rba
```

1.20 S

1.20.1 SATTACH

Use the SATTACH command to attach a specified table from a foreign database to a connected R:BASE database.

```

SATTACH [TEMPORARY] tblname [AS tablealias] ...
... [USING [ONLY | ALL] collist] [ALIAS aliaslist] [WHERE clause]

```

Options

ALIAS AliasList

To specify alias names for columns. The alias list is separated by commas. Only the changed column names can be specified in the alias list, with other column name to be retained as is left blank.

AS tablealias

Specifies an alias, or temporary name, for the foreign table. A table alias is sometimes required when attaching foreign data sources that do not follow the same table name restrictions as R:BASE.

tblname

Specifies the table in the foreign database to attach.

TEMPORARY

Allows you to create a temporary name with the SATTACH command. The temporary tables will disappear when the database is disconnected. **NOTE:** Any changes made to the temporary table will not be reflected upon the original SQL data source.

USING ALL

Specifies all columns that uniquely identify the rows in an attached table if no primary or unique keys are defined. With USING ALL, the [QUALCOLS](#) setting is ignored to determine the number of columns to identify the rows.

When performing a direct [UPDATE](#) to the foreign table, the USING ALL approach is the slowest in processing, which means that to qualify a row for updating, all of the column values must match. Rather, if a primary key exists, specify the column with USING *PrimaryKeyColName* instead, as this way, only the primary key value must match (which is all that should be needed). When updating a row on the foreign table, R:BASE must count how many rows match that row, and there should only be one matching row.

USING collist

If the foreign table has no primary or unique key, specify the column(s) that uniquely identify the rows in the table. The primary key should be specified as the *collist* value. The *collist* is not limited to a single column, but the more columns that are specified, then those column values must also match.

USING ONLY collist

Specifies that only the columns listed will be attached. When using the ONLY option the word "ONLY" must immediately follow the word "USING". R:BASE will determine the key columns by querying the special columns of source table, to find primary key or unique key columns. If the query fails, then all columns will be used.

WHERE clause

Limits rows of data. For more information, see the [WHERE](#) Clause.

About the SATTACH Command

Before you can attach a foreign data source table, an R:BASE database must be connected. Also, your workstation must be connected to the data source.

If you use SATTACH without the *tblname* option, R:BASE displays the "**Attach Table(s)**" dialog box with names of tables in the data source. You can then select a table to attach.

When attaching external tables by selecting "Utilities" > "Attach SQL Database Tables" from the menu bar, or using the SATTACH command (without the "USING ALL/collist" keywords), the [QUALCOLS](#) setting is used to assign what columns uniquely identifies a row. If a primary key or unique key was not found for the table being SATTACHed, and the USING collist clause was not used to specify what columns uniquely identifies a row, then R:BASE assigns primary and unique key qualkeys for the attached table. R:BASE assigns a set of columns to identify the rows starting with the first column of the table. The number of columns used is limited by the value for QUALCOLS. The (CVAL('QUALKEYS')) function may be used to capture the columns assigned as a QualKeys for the current database. The (CVAL('QUALKEY TABLES')) function may be used to capture the tables assigned with QualKey columns.

After you have exited R:BASE or disconnected the database to which the foreign table is attached, you do not need to reconnect to the table's data source when you open the database again. The data source is connected when you use the attached table. The data source table remains attached until you detach it with the [SDETACH](#) command, or use the **Utilities: Detach SQL Database Table** menu option.

When a foreign table is attached, R:BASE writes a table description that identifies the table as a data source table and names its data source. Use the [LIST TABLES](#) *tblname* command to review table descriptions.

Notes:

- SATTACH requires an owner password if one has been defined, or permission to create tables.
- When you attach a foreign table, R:BASE only includes the columns with legal names. For example, R:BASE does not include columns that have spaces in the name, or column names that exceed the

character limit for the R:BASE version installed. Table and column names are limited to 128 characters.

- When you attach a foreign table and select the columns that uniquely identify its rows in the "Select Column Set" dialog box, do not select columns that have LOB data types, as unpredictable results might occur.
- When running applications that connect to foreign data sources, you should always disconnect from the R:BASE database before running the application again.
- When using the keyword "ONLY", to limit the columns attached with a table, the ability to SATTACH temporary tables and column alias names is supported.

Notes for [Alias] Parameters:

- Syntax has been extended to specify only the changed columns. For example, if you only need to alias the second column out of four columns you can use ... *ALIAS* ,,location,,
- Any missing alias names will use the default name.
- If there is a conflicting column name, a warning will be displayed you will be prompted for a new column name.
- If the name conflicts with another name then you get the error message first explaining the conflict, then the dialog box.
- If no qualkey is specified, automatic qualkeys will be assigned based on information from the ODBC source

Examples:

Example 01:

Attaches a foreign data source table using an alias table name

```
SATTACH CustomerDetails AS tCustomerDetails
```

Example 02:

Attaches a foreign data source table using an alias table name, and specifies the unique column name for the source table

```
SATTACH CustomerDetails AS tCustomerDetails USING CustomerID
```

Example 03:

Attaches a foreign data source table using alias names for the table and columns, and specifies the unique column name for the source table

```
SATTACH Orders AS tOrders USING +
OrderID ALIAS +
OrderID, +
CustomerID, +
EmployeeID, +
OrderDate, +
RequiredDate, +
ShippedDate, +
ShipVia, +
tFreight, +
ShipName, +
ShipAddress, +
ShipCity, +
ShipRegion, +
ShipPostalCode, +
ShipCountry
```

Example 04:

Attaches a foreign data source table using alias names for the table and only the Freight column, and specifies the unique column name for the source table

```
SATTACH Orders AS tOrders USING +
OrderID ALIAS ,,,,,,,tFreight,,,,,
```

Example 05:

Attaches a foreign data source table using an alias name for the table which contains spaces, and specifies only two columns to be included. The table `Order Details` is surrounded by [IDQUOTES](#).

```
SATTACH `Order Details` AS tOrderDetails USING ONLY +
OrderID, ProductID
```

Example 06:

Attaches a foreign data source table using an alias name for the table, and specifies a WHERE clause for limited results

```
SATTACH Artists AS tArtists WHERE ALastName = 'Ford'
```

1.20.2 SAVEROW

Use the SAVEROW command in an entry/exit procedure (EEP) to save a row.

SAVEROW

About the SAVEROW Command

Use the SAVEROW command to save changes to the row of data currently being edited or entered in a form.

SAVEROW is not the same as **Save Changes** option on the **Forms** menu used in R:BASE for DOS, which saves the edits to the data then displays the next row. To perform the same functions as this menu option, use the SAVEROW command followed by the [NEXTROW](#) command.

The SAVEROW command, used in an exit procedure, leaves you in the field from which it was called. To move to the next field, add the command [SKIP 1](#) to the EEP.

1.20.3 SCONNECT

Use the SCONNECT command to connect R:BASE to a foreign data source.

```
SCONNECT [datasource] IDENTIFIED BY userid [password]

DSN-Less Connection:

SCONNECT ' ;driver=<Complete ODBC Driver Name>;dbq=<database path and name>' ...
... IDENTIFIED BY userid [password]
```

Options

datasource

Specifies the name of the data source name (DSN) that contains the table to access.

IDENTIFIED BY userid

Specifies the user account name for the data source. You can use " (two single quotes) in this position if there is no User ID.

password

Specifies the password for the data source. You can use " (two single quotes) in this position if there is no password.

About the SCONNECT Command

To access an R:BASE database, the R:BASE or Oterro ODBC driver must be installed. The R:BASE/Oterro ODBC driver version must match the version of the database files that are being connected.

Omitting the *datasource* option will display a dialog box, listing data sources from which to choose, and continues with prompts for a *userid* and *password*.

Use the [SATTACH](#) command to attach tables to a database. Added data source tables, or SERVER tables, within an R:BASE database will remain defined as part of the database structure, and will appear in the Database Explorer table list. To disconnect a DSN connection, use the same syntax with the [SDISCONNECT](#) command.

About the DSN-Less Connection

A data source name (DSN) is a data structure that contains the information about a specific database that an Open Database Connectivity (ODBC) driver needs in order to connect to it. Included in the DSN, which resides either in the registry or as a separate text file, is information such as the name, directory and driver of the database, and, depending on the type of DSN, the ID and password of the user. The developer creates a separate DSN for each database. To connect to a particular database, the developer specifies its DSN within a program. In contrast, DSN-less connections require that all the necessary information be specified within the command. DSN-Less connection requires no server setup, just a carefully constructed connection string.

There are three kinds of DSN: user DSNs (sometimes called machine DSNs); system DSNs; and file DSNs. User and system DSNs are specific to a particular computer, and store DSN information in the registry. A user DSN allows database access for a single user on a single computer, and a system DSN for any user of a particular computer. A file DSN contains the relevant information within a text file with a .DSN file extension, and can be shared by users of different computers who have the same drivers installed.

DSN-less connections demand that you know the name of the file (e.g. file based databases like R:BASE) or the address of the data server (SQL Server for example).

Armed with appropriate information you could open a data source without a DSN. Normally on the SCONNECT command you specify the DSN you want to use. ODBC looks up this DSN and determines the driver to use and what connection it needs. With this method, R:BASE allows the ability to specify the ODBC driver instead of a DSN.

About the Connection String

The first character in the connection string is a semi-colon. This is the flag which states the string is not specifying a DSN.

The value after "driver" is the actual name of the driver as defined in the ODBC Data Source Administrator. Make sure to spell and space the driver name exactly as it is defined in ODBC Administrator interface under the "Drivers" tab.

The database name and exact path is specified in the "dbq" portion of the string.

The entire string must be surrounded by the database [QUOTE](#) character. Single quotes are used in the examples below.

Examples

Example 01 (Using R:BASE 11 ODBC Driver):

```
SCONNECT ';'driver=R:BASE 11 Database Driver (*.RX1);dbq=d:\SampleData\RRBYW20'
```

Example 02 (Using Oterro 11 ODBC Driver):

```
SCONNECT ';'driver=Oterro 11 Database Driver (*.RX1);dbq=d:\SampleData\RRBYW20'
```

Example 03 (To use an Access database in "dsnless" mode):

Uses a database called db1.mdb in the "My Documents" folder. The connection can specify additional items like user id with the "UID=" parameter and a password with the "PWD=" parameter.

```
SCONNECT ' ;Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\Documents and
Settings\Administrator\My Documents\db1.mdb; '
```

Example 04 (To use a SQL Server database):

The server and database are specified along with the uid and pwd.

```
SCONNECT ' ;driver={SQL
Server};server=corpseadb0d;uid=my_user_name;pwd=my_pw;database=JohnDoe; '
```

1.20.4 SCREEN RESTORE

Use the SCREEN RESTORE command in forms when you want to redisplay the screen after returning from an entry/exit procedure (EEP).



Options

OFF

Set SCREEN RESTORE off to prevent the original form from displaying any changes after an entry/exit procedure executes in a form.

ON

Set SCREEN RESTORE on to redisplay the original form after an entry/exit procedure is executed in a form.

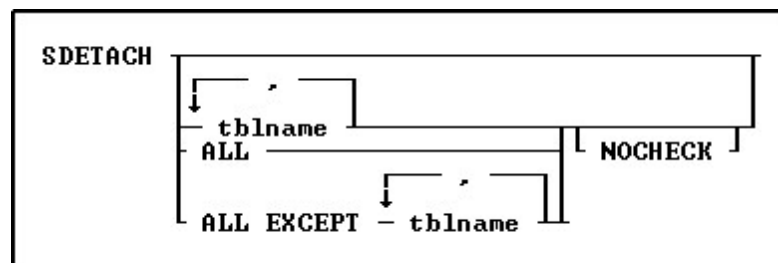
About the SCREEN RESTORE Command

The SCREEN RESTORE ON command, which is the default, will redisplay the original screen. The SCREEN RESTORE OFF command does not redisplay the original screen, allowing for screen I/O during execution of the EEP to remain when control returns to the form.

Note: The SCREEN RESTORE command is specific to R:BASE for DOS.

1.20.5 SDETACH

Use the SDETACH command to remove a foreign data source table from a connected R:BASE database.



Options

' Indicates that this part of the command is repeatable.

ALL

Specifies all tables.

ALL EXCEPT tblname

Specifies all tables except those specified.

NOCHECK

Eliminates the confirmation message.

tblname

Specifies the table to detach.

For information on how to attach foreign data source tables, see [SATTACH](#).

Examples:

Example 01:

Detaches the tOrderDetails foreign table, without displaying the confirmation dialog

```
SDETACH tOrderDetails NOCHECK
```

Example 02:

Detaches all foreign data source tables except ServerFreight and ServerShipping

```
SDETACH ALL EXCEPT ServerFreight, ServerShipping
```

Example 03:

Detaches the dbo.Orders foreign table, where [IDQUOTES](#) are used as the name contains a period

```
SDETACH `dbo.Orders`
```

1.20.6 SDISCONNECT

Use the SDISCONNECT command to disconnect a foreign data source from an R:BASE database.

SDISCONNECT datasource

Options**datasource**

Specifies the data source to disconnect.

About the SDISCONNECT Command

If you omit the *datasource* option, a dialog box opens listing data sources from which to choose.

To disconnect the DSN-less connection, use the exact same data source syntax as the [SCONNECT](#) command, only with SDISCONNECT.

Examples:

Example 01:

```
SDISCONNECT ' ;driver=R:BASE 11 Database Driver (*.RX1);dbq=d:\SampleData\RRBYW20 '
```

Example 02:

```
SDISCONNECT ' ;driver=Oterro 11 Database Driver (*.RX1);dbq=d:\SampleData\RRBYW20 '
```

Example 03:

```
SDISCONNECT ' ;Driver={Microsoft Access Driver (*.mdb)};DBQ=c:\Documents and
Settings\Administrator\My Documents\db1.mdb; '
```

Example 04:

```
SDISCONNECT
' ;driver={SQLServer};server=corpseadb0d;uid=my_user_name;pwd=my_pw;database=JohnDoe; '
```

See also: [DSN-less connections](#)

1.20.7 SELECT

Use the SELECT command to display rows of data from a table or view. To display the data in the order you want, modify the SELECT command by using various clauses.

The SELECT command is a very powerful data retrieval command. By learning this command, and all of its parts you can greatly enhance your ability to work with any other R:BASE command that uses those same portions. For example, learning to use a WHERE clause with SELECT will help you work with WHERE clauses on other commands.

You can use the SELECT command to do the following:

- Display rows of information from a table or view
- Extract information from a table or view by using a [sub-SELECT](#) command (a nested SELECT statement) in a WHERE command
- Extract information from a table or view by using a SELECT clause in another command

A SELECT command is essentially a process of elimination. A SELECT command can contain a number of clauses (two are required), each of which begins with a keyword, such as [FROM](#) or [WHERE](#).

The diagram below shows the different clauses in a SELECT command.

```
SELECT... TOP... SELECT functions... INTO...
FROM... EXCEPT... LIMIT... INNER JOIN...
OUTER JOIN... WHERE... Sub-SELECT... AS...
GROUP BY... HAVING... ORDER BY...
UNION... HTML...
```

Each of the SELECT clauses has a specific purpose for determining what data you want. The operators are processed in the order in which they appear in the preceding diagram.

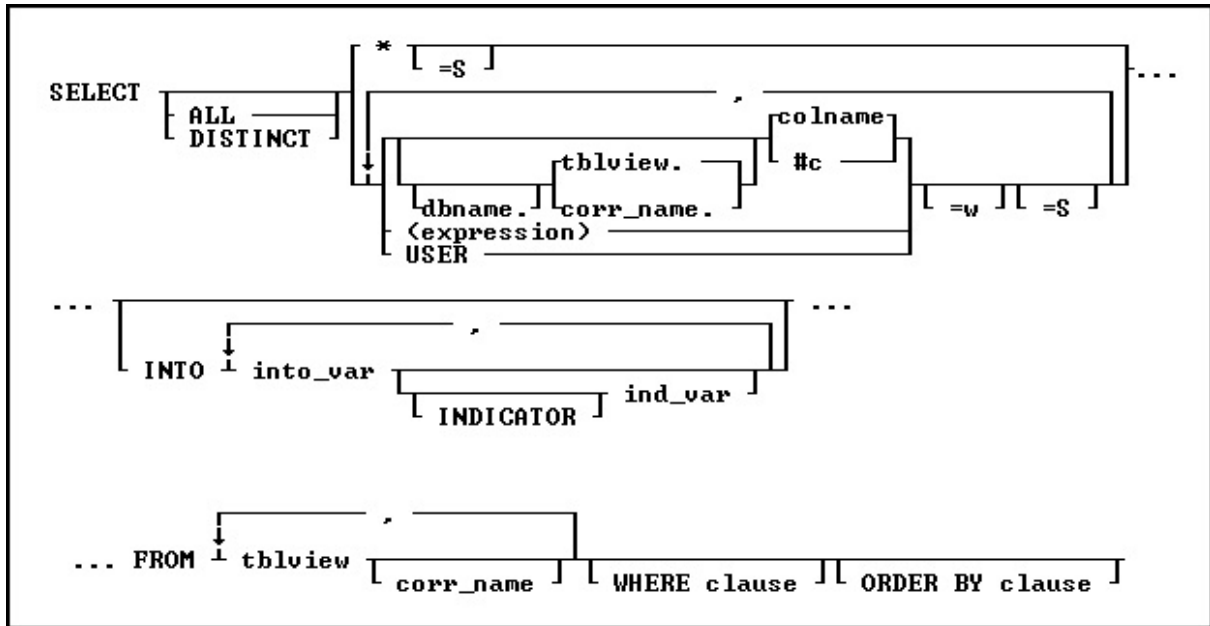
Note:

- Many of the SELECT clauses use the same options, such as *expression* or *colname*. These common options are described only once in "SELECT Command Clause" below.

SELECT Command Clause

The required SELECT command clause specifies which columns to include. You can:

- Select all columns by entering SELECT with an asterisk.
- Name the columns you want to select.
- Use expressions and SELECT functions to perform calculations whose results will also appear as a column in the final result.
- Select the column or expression values and load them into variables.

Syntax:**Options**

Specifies all columns.

,

Indicates that this part of the command is repeatable.

ALL

Specifies all rows returned by the other clauses.

#c

Specifies a column, where **#c** is the column number shown in the output of the [LIST TABLES](#) command. You can enter a table or correlation name before the **#c**.

colname

Specifies a column name. In a command, you can enter **#c**, where **#c** is the column number shown when the columns are listed with the **LIST TABLES** command. In an SQL command, a column name can be preceded by a table or correlation name and a period (**tblname.colname**). You can enter **tblname.*** to specify all columns in the table.

corr_name

Correlation name. A nickname or alias for a table or view name. Use **corr_name** to refer to the same table twice within the command, or to explicitly specify a column in more than one table. The correlation name must be at least two characters.

dbname

Currently connected database name, plus the drive and directory if the database is not on the current directory. It has the form **D:\PATHNAME/DBNAME** where **D:** is the optional drive letter, **/PATHNAME** is the optional directory path, and **/DBNAME** is the database name.

DISTINCT

Eliminates duplicate rows from the resulting data set.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as **#date**, **#time**, and **#pi**.

FROM

Lists the tables from which data is to be displayed.

ind_var

Specifies a variable result indicator to be used with an [INTO](#) clause in a SELECT command. This variable stores the status of the variable: non-null (0) or null (-1).

INDICATOR

Indicates the following variable is an indicator variable, which is used to indicate if a null value is retrieved.

INTO

Selects information directly from a table and puts it into variables. You must include a WHERE clause so the SELECT command finds only one row.

into_var

Specifies a variable whose value is assigned with an INTO clause in a SELECT command.

ORDER BY clause

Sorts rows of data. See [ORDER BY](#).

=S

Calculates the sum of a column that has CURRENCY, DOUBLE, INTEGER, NUMERIC, or REAL data type values, or the results of an expression using CURRENCY, DOUBLE, INTEGER, NUMERIC, or REAL data type values.

tblview

Specifies a table or view name.

USER

Retrieves the current user as a constant.

=w

Specifies a display width.

WHERE clause

Limits rows of data. See [WHERE](#).

Examples

The following command selects the company name and ID for companies in Washington state:

```
SELECT custid, company FROM customer +
WHERE custstate = 'WA' ORDER BY company
```

custid company

```
122    Data Solutions
119    Datacrafters Infosystems
130    MIS by Design
114    Softech Database Design
```

The following command selects the first, fourth, and fifth columns from the Employee table:

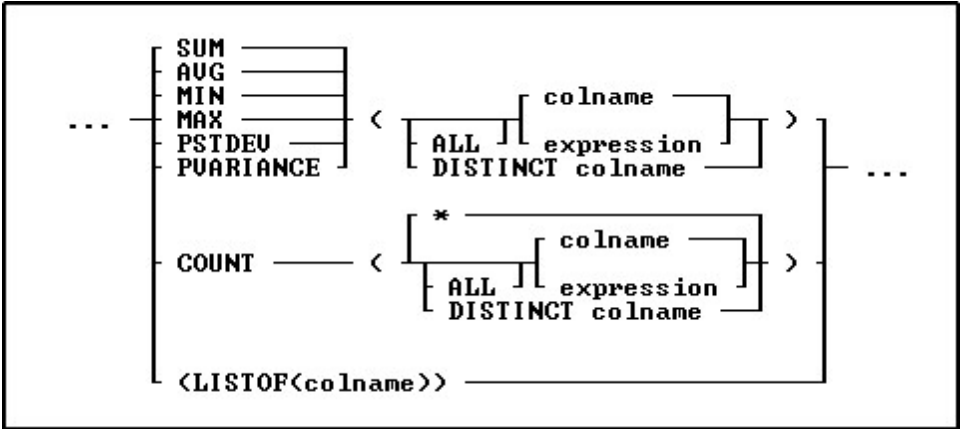
```
R>SELECT #1, #4, #5 FROM Employee
```

EmpID	EmpFName	EmpLName
101	Arielle	Wilson
102	Ernest	Hernandez
103	Mary	Simpson
104	Peter	Coffin
105	John	Smith
106	Darnell	Williams

107	John	Chow
108	Jane	Sullivan
109	John	Smith
110	Sam	Donald
111	Joe	Donohoe
115	Craig	Alan

1.20.7.1 SELECT Functions

This clause, determines which columns to include.



Options

*

Specifies all rows, without regard to the values of the columns in any of the rows. From a performance standpoint, using "COUNT(colname)" would be slightly slower because in addition to fetching each row, it has to look at the column value in that row.

AVG

Computes the numeric average. R:BASE rounds averages of INTEGER values to the nearest integer value and CURRENCY values to their nearest unit.

COUNT

Determines how many non-null entries there are for a particular column item. The returned count value could be smaller than the table row count if there are null values.

DISTINCT

Eliminates duplicate rows from the calculation. The DISTINCT keyword is only needed to be specified for the first column in order to be applied to all columns in a query.

LISTOF

Creates a text string of the values separated by the current comma delimiter character. The text string is limited to 30,000 characters. As LISTOF is an aggregate-style function, any sorting ([ORDER BY](#)) needed for the listed results must be performed at a lower level, in which a view may be created to perform the sorting first.

The LISTOF function can be used with the "SELECT ... INTO ..." to populate a variable with a list of values which can then be used in a [CHOOSE](#) command with the #LIST option. It can also be used in forms, reports, or labels to look up values from multiple rows.

MAX

Selects the maximum numeric, time, date, or alphabetic value in a column.

MIN

Selects the minimum numeric, time, date, or alphabetic value in a column.

PSTDEV

Calculates population standard deviation.

PVARIANCE

Determines population variance.

SUM

Computes the numeric sum.

About SELECT Functions

A SELECT function can be used to provide summary data about a group of rows in a table or for all rows in a table. These functions may only be used with the GROUP BY clause or when only SELECT functions are specified.

Using MIN and MAX functions for NOTE data type

Selecting aggregate functions, such as MIN and MAX, requires that R:BASE keeps an accumulator and choose to only use the first 2,000 characters for NOTE/WIDENOTE values. If you sort on NOTE/WIDENOTE fields, the sort will be based on the first 80 characters only.

Examples

The COUNT function works in two different ways, depending on its argument. COUNT(*) counts all rows in a table, but COUNT(*colname*) counts only rows in which the value in the specified column is not null. For example:

```
SELECT emptitle, COUNT(*), COUNT(emptitle) +
FROM employee GROUP BY emptitle
```

The final result contains both forms of the COUNT function.

emptitle	COUNT(*)	COUNT(emptitle)
Manager	2	2
Representative	3	3
Sales Clerk	1	1
-0-	2	0

If you wanted to compute the difference between each employee's average sales and the average sales for all employees, you would first use a SELECT function to calculate the average for all employees and store the result in a variable. For more information, see [INTO](#).

The following command assigns the value \$71,571.88 to the *vaverage* variable.

```
SELECT AVG(netamount) INTO vaverage FROM transmaster
```

Next, you can use the variable and another SELECT function in an expression to calculate the difference for each employee, and display the average net amount for each employee.

```
SELECT empid, AVG(netamount), +
    (.vaverage - (AVG(netamount)))=30 +
FROM transmaster GROUP BY empid
```

empid	AVG (netamount)	(.vaverage - AVG(netamount))
102	\$64,510.00	\$7,061.88
129	\$69,555.00	\$2,016.88
131	\$118,000.00	-\$46,428.12
133	\$44,380.00	\$27,191.88
160	\$114,850.00	-\$43,278.12
165	\$14,685.00	\$56,886.88
167	\$3,830.00	\$67,741.88

Examples Using the LISTOF Function

```
SELECT (LISTOF(ColumnName)) INTO VariableName INDICATOR IndVar +
FROM TableView WHERE ...
```

In a Form, Report or Label Expression:

```
VariableName = (LISTOF(ColumnName)) IN LookUpTableView WHERE +
KeyColumn = KeyColumn
```

Example 01:

```
SET VAR vValueList TEXT = NULL
SELECT (LISTOF(ColumnName)) INTO vValueList INDIC IvValueList +
FROM TableName WHERE ...
```

The variable vValueList will be a text string of the values separated by the current comma delimiter character.

If you would like to add a single space after each value, then:

```
SET VAR vValueList TEXT = NULL
SELECT (SRPL(LISTOF(ColumnName),', ',' ',0)) INTO +
vValueList INDIC IvValueList FROM TableName WHERE ...
```

Notice the additional space after the comma in SRPL function's replace string value.

If you would like to use a carriage return after each value, then:

```
SET VAR vValueList TEXT = NULL
SELECT (SRPL(LISTOF(ColumnName),', ',' ',(CHAR(10)),0)) INTO +
vValueList INDIC IvValueList FROM TableName WHERE ...
```

Example 02:

```
CONNECT Concomp IDENTIFIED BY NONE
SET CAPTION ' '
SET AUTODROP OFF
SET RBGSize CENTER CENTER 800 600
SET VAR vLines INTEGER = 0
SET VAR vValueList TEXT = NULL
SET VAR vLastName TEXT = NULL
SET VAR vTitle TEXT = 'List Created Using LISTOF Function'
SET VAR vCaption TEXT = 'Using #LIST Options in CHOOSE Command!'
CLS
PAUSE 3 USING 'Collecting Values ...' CAPTION .vCaption AT 16 30
SELECT (COUNT(*)), (LISTOF(EmplName)) INTO +
vLines INDIC IvLines, vValueList INDIC IvValueList FROM Employee
IF vLines > 18 THEN
    SET VAR vLines = 18
ENDIF
CLS
CHOOSE vLastname FROM #LIST .vValueList AT 6 30 +
TITLE .vTitle CAPTION .vCaption LINES .vLines FORMATTED
IF vLastName IS NULL OR vLastName = '[Esc]' THEN
    GOTO Done
ELSE
```

```

    CLEAR ALL VAR EXCEPT vLastName
ENDIF
-- Do what you have to do here ...
LABEL Done
CLS
CLEAR ALL VAR
QUIT TO MainMenu.RMD
RETURN

```

Example 03:

The following example retrieves a list of DISTINCT values from the *InvoiceHeader* table.

```

SET VAR vValueList TEXT = NULL
SELECT (LISTOF(DISTINCT CustID)) INTO vValueList INDIC IvValueList FROM InvoiceHeader

100,101,102,103,104,105,106,10
7,108,109,110,111,112,113,114,
115,116,117,118,119,120,121,12
2,123,124,125,126,127,128,129

```

Example 04:

The following example retrieves a list of DISTINCT state and zip code values from the *Employee* table.

```

SELECT DISTINCT EmpState,EmpZip, FROM Employee
EmpState EmpZip
-----
PA      15668-1854
WA      98004
WA      98072
WA      98101
WA      98102
WA      98103
WA      98115

```

1.20.7.2 SELECT Analytical Functions

The LAG and LEAD analytical functions can be use to specify more than one row in a table at a time without having to join the table to itself.

```

... [ LAG ] < [ colname ] [ , offset [ , default ] ] > [ AS alias ] ...
... [ LEAD ] < [ colname ] [ , offset [ , default ] ] > [ AS alias ] ...

... [ OVER < [ PARTITION BY colname ] ORDER BY clause > ] ...

```

Options**AS alias**

Specifies an alias name for the column/expression.

colname

Specifies a column name.

default

Specifies the value that is returned if the offset goes out of the bounds of the table. If this optional parameter is omitted, the default is NULL. The default must be a value that can be evaluated from looking at data in a single row.

expression

Determines a value using a text or arithmetic formula. The expression must be a value that can be evaluated from looking at data in a single row. Lookup expressions are not supported.

LAG

Specifies the LAG analytic to query more than one row in a table at a time returning values from a previous row in the table.

LEAD

Specifies the LEAD analytic to query more than one row in a table at a time returning values from a next row in the table.

offset

Specifies the physical offset from the current row in the table. If this optional parameter is omitted, the default is 1. The offset must be a value that can be evaluated from looking at data in a single row.

ORDER BY clause

Sorts rows of data, and supports standard [ORDER BY](#) clauses with ASC and DESC directions. When used, there should be no other ORDER BY clauses in the command.

OVER

Provides an optional specification for grouping and sorting. By specifying the OVER syntax, the command must also contain an ORDER BY clause. For the overall SELECT, the PARTITION BY and ORDER BY will determine the sequencing of the output.

PARTITION BY

Specifies optional grouping of the data where the LEAD or LAG function is restricted to be computed within each group. It is very similar to how [GROUP BY](#) clauses work for functions like [MIN](#) and [MAX](#). For example, a table containing several students with several test scores for each student, a PARTITION BY *student* will mean that a given student's LEAD or LAG functions will come only from other rows that belong to that same student.

Note:

- LEAD and LAG functions are "SELECT only" type expressions and cannot be used in [SET VAR](#) commands.

Examples:

Example 01:

The following example uses the LAG function to compare prior sales between California companies. The PARTITION BY clause is specified to divide the rows in the result set by company. The ORDER BY clause in the OVER clause orders the rows in each partition. Notice that because there is no lag value available for the first row, and where a single purchases exists, the default of zero (\$0.00) is returned.

```
SELECT BillToCompany, TransDate, NetAmount, +
      LAG(NetAmount,1,0) AS PrevAmount +
      OVER (PARTITION BY BillToCompany ORDER BY TransDate) +
FROM InvoiceHeader WHERE BillToState = 'CA'
```

BillToCompany	TransDate	NetAmount	PrevAmount
Compumasters Computer Supply	01/17/2018	\$7,775.00	\$0.00
Compumasters Computer Supply	03/01/2018	\$3,080.00	\$7,775.00
Compumasters Computer Supply	06/12/2018	\$8,955.00	\$3,080.00
Compumasters Computer Supply	10/12/2018	\$4,308.00	\$8,955.00
Compumasters Computer Supply	11/13/2018	\$3,512.00	\$4,308.00

Compumasters Computer Supply	12/01/2018	\$1,615.50	\$3,512.00
Compumasters Computer Supply	12/22/2018	\$20,852.50	\$1,615.50
Industrial Concepts Inc.	05/01/2018	\$4,477.50	\$0.00
Johnson Technologies	05/02/2018	\$1,881.00	\$0.00
Open Systems I/O	07/06/2018	\$2,390.00	\$0.00
PC Consultation And Design	07/07/2018	\$9,450.00	\$0.00
PC Consultation And Design	07/15/2018	\$2,772.00	\$9,450.00

Example 02:

The following example uses the LEAD function to compare employee sales variations for the first quarter. The ORDER BY clause in the OVER clause orders the last names in each date partition.

```
SELECT TransDate, (EmpFName&EmpLName)=20, NetAmount, +
  LEAD(NetAmount,1,0) AS NextAmount +
  OVER (PARTITION BY TransDate ORDER BY EmpLName) +
  FROM SalesByEmployee WHERE (IMON(TransDate)) < 4
```

TransDate	(EmpFName&EmpLName)	NetAmount	NextAmount
01/05/2018	Peter Coffin	\$17,560.00	\$13,941.00
01/05/2018	Ernest Hernandez	\$13,941.00	\$0.00
01/06/2018	John Chow	\$895.50	\$0.00
01/14/2018	Ernest Hernandez	\$13,572.00	\$0.00
01/15/2018	Joe Donohoe	\$19,755.00	\$0.00
01/17/2018	Peter Coffin	\$7,775.00	\$0.00
01/22/2018	Joe Donohoe	\$4,508.75	\$1,975.50
01/22/2018	Ernest Hernandez	\$1,975.50	\$0.00
01/23/2018	Sam Donald	\$8,616.00	\$15,551.00
01/23/2018	Ernest Hernandez	\$15,551.00	\$4,972.50
01/23/2018	John Smith	\$4,972.50	\$0.00
02/01/2018	Peter Coffin	\$1,672.00	\$16,155.00
02/01/2018	Sam Donald	\$16,155.00	\$0.00
02/26/2018	Peter Coffin	\$3,231.00	\$0.00
03/01/2018	Sam Donald	\$3,080.00	\$7,024.00
03/01/2018	John Smith	\$7,024.00	\$0.00
03/09/2018	Joe Donohoe	\$7,182.00	\$0.00
03/10/2018	John Smith	\$2,772.00	\$0.00
03/20/2018	Mary Simpson	\$4,158.00	\$0.00
03/21/2018	John Smith	\$7,392.00	\$0.00
03/23/2018	Darnell Williams	\$2,151.00	\$0.00

1.20.7.3 TOP

This parameter includes the support to specify the TOP n qualifier for the [SELECT](#) command. The TOP n will retrieve the top number of records from the table.

```
SELECT TOP n [ ALL collist ] FROM tablename ...
```

The "TOP n" goes between the word SELECT and the column list for the command.

n

Specifies the number of records to retrieve

collist

Specifies a list of one or more column names, separated by a comma (or the current delimiter)

Examples:

To show the top 5 bonuses where the bonus is under \$500 from the SalesBonus table in the ConComp sample database:

```
SELECT TOP 5 EmpID,Bonus FROM SalesBonus WHERE Bonus < 500 ORDER BY Bonus=DESC
```

EmpID	Bonus
131	\$456.75
131	\$326.25
102	\$175.00
131	\$157.50
129	\$153.60

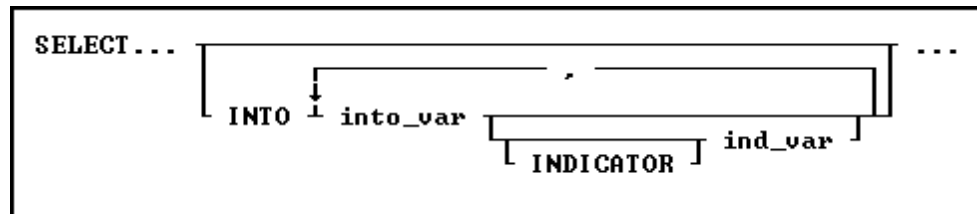
To show the three companies with the largest purchases for the RRBYW20 sample database:

```
SELECT custid,company FROM customer WHERE custid IN +
(SELECT custid FROM invoiceheader WHERE netamount IN +
(SELECT TOP 3 netamount FROM invoiceheader))
```

custid	company
121	Bytes & Words
124	Barton and Associates
127	RAM Data Systems, Inc.

1.20.7.4 INTO

If the result consists of one row, this clause loads the data into one or more variables, one for each column value in the result.



Options

ind_var

Stores an INTEGER value (-1 or 0) that indicates whether the preceding *into_var* received a null value or a non-null value; this is an optional indicator variable. If you omit indicator variables, R:BASE displays a message and assigns a negative integer to SQLCODE if it encounters a null value. The command continues to process rows.

INDICATOR

Indicates the following variable is an indicator variable, which is used to indicate if a null value is retrieved.

into_var

Assigns the result associated with a column, expression, or function named in the command clause to the corresponding variable named in the INTO clause. The number of items or variables named in the command and INTO clauses, as well as their data types, must be the same.

About the SELECT INTO command

This optional clause loads the results of a SELECT command into variables, but does not display the results on screen.

An INTO clause loads the resulting value of each column, expression, or function included in the command clause into a variable. If previous clauses have returned more than one row, the values assigned to the variables are unpredictable. You should make sure you are returning only one row. Either test the results before using an INTO clause or check the value of the variable *sqlcode* after executing the command. If the clause is successful, *sqlcode* is 0.

Comments

The INTO clause must have a corresponding variable for every item in the command clause; values are assigned to variables in the order of items in the command clause. The data type of each command clause item and its corresponding *into_var* must be compatible. For example:

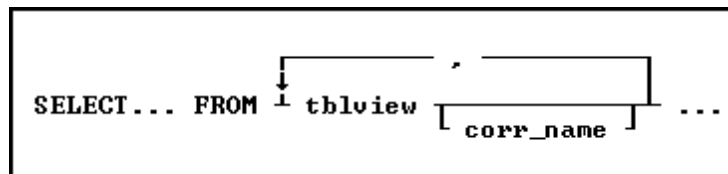
```
SELECT MAX(listprice), MIN(listprice) +
      INTO vmaxprice INDICATOR vind_max, +
          vminprice INDICATOR vind_min +
      FROM product
```

The MAX and MIN functions assign the value \$3,100.00 to the variable *vmaxprice* and \$1,900.00 to *vminprice*. These values are the maximum and minimum values for the *listprice* column in the *product* table. Since both functions returned values, the value of both indicator variables is 0. Also, since only SELECT functions are specified, a [GROUP BY](#) clause is not required.

If you select and load a value into an undefined numeric variable, that variable acquires the precision and scale of the column from which the value is selected.

1.20.7.5 FROM

Starting with all the tables, views, rows, and columns in the database, this clause specifies one or more tables or views from which you want data.



Options

,

Indicates that this part of the command is repeatable.

corr_name

A correlation name is an alias or nickname for a table. It lets you refer to the same table twice in one command, use a shorter name, and explicitly refer to a column when referring to the same column if that column appears in more than one table. The correlation name must be at least two characters.

tblview

A table or view containing one or more columns named in the command clause.

About the FROM Clause

The FROM clause names one or more tables and/or views from which the information is used in a SELECT command or other command. It is one of the two **REQUIRED** portions of a SELECT statement. The other required portion being the column listing. Some other commands that may use a FROM clause include [TALLY](#), [COMPUTE](#) and [CHOOSE](#).

Examples

The following command selects all columns from the *transmaster* table in the R:BASE sample database, *concomp*.

```
SELECT * FROM transmaster
```

The result of this command appears in the following table. The *transid* column is the primary key for this table; that is, *transid* contains a unique value for each row in the table. Columns that are not primary keys can have the same value in more than one row. The result shown here is used in the discussions of other SELECT clauses later in this section.

transid	custid	empid	transdate	netamount	freight
4760	100	133	01/02/94	\$32,400.00	\$324.00
4780	105	160	01/08/94	\$9,500.00	\$95.00
4790	104	129	01/09/94	\$6,400.00	\$64.00
4795	101	102	01/11/94	\$176,000.00	\$1,760.00
4800	105	160	02/22/94	\$194,750.00	\$1,947.50
4865	102	129	02/22/94	\$34,125.00	\$341.25
4970	103	131	02/23/94	\$152,250.00	\$1,522.50
4975	101	102	02/26/94	\$87,500.00	\$875.00
4980	101	102	02/27/94	\$22,500.00	\$225.00
5000	101	102	02/28/94	\$40,500.00	\$405.00
5010	107	131	03/02/94	\$108,750.00	\$1,087.50
5015	103	131	03/05/94	\$80,500.00	\$805.00
5050	104	129	03/06/94	\$56,250.00	\$562.50
5060	101	102	03/07/94	\$57,500.00	\$575.00
5065	106	160	03/13/94	\$140,300.00	\$1,403.00
5070	104	129	03/14/94	\$95,500.00	\$955.00
5075	102	129	03/15/94	\$155,500.00	\$1,555.00
5080	100	133	03/19/94	\$88,000.00	\$880.00
5085	107	131	03/18/94	\$130,500.00	\$1,305.00
5045	100	102	09/26/94	\$3,060.00	\$30.60
5046	101	165	09/27/94	\$3,060.00	\$30.60
5047	102	167	09/27/94	\$3,830.00	\$38.30
5048	103	133	-0-	\$12,740.00	\$127.40
5049	102	165	04/21/94	\$26,310.00	\$263.10

When a column appears in more than one table, enter the table name and a period preceding each column name to specify the column you want. For example:

```
SELECT transmaster.transid, transmaster.netamount,+
transdetail.model FROM transmaster, transdetail +
WHERE transmaster.transid = transdetail.transid
```

Or, you can assign a correlation name to a table. The following command is equivalent to the previous example:

```
SELECT t1.transid, t1.netamount, t2.model +
FROM transmaster t1, transdetail t2 +
WHERE t1.transid = t2.transid
```

In this SELECT command, the FROM clause assigns correlation names to the *transmaster* and *transdetail* tables. Because the *transid* column appears in both tables, the correlation names, *t1* and *t2*, clarify which table each column is from.

Because R:BASE processes the FROM clause first, you must use correlation names, if you have assigned them, throughout the SELECT command.

1.20.7.6 EXCEPT

The EXCEPT clause allows for specified columns to be excluded from a [SELECT](#) query.

```
SELECT [ ALL ] FROM tablename EXCEPT [ colname ] ...
```

The EXCEPT clause is another clause you can have in a SELECT command like [WHERE](#) and [ORDER BY](#).

SELECT ... EXCEPT selects data from all records in the table, except for columns in exclusion list. When using SELECT ... EXCEPT there can only be one table in the [FROM](#) clause.

The EXCEPT clause is helpful with tables which have many columns, to exclude columns from the result set. EXCEPT saves time to avoid typing a long list of column names in a SELECT statement.

Examples

The below selects all columns from the *Customer* table, except *CustURL* and *CustEMail*

```
SELECT ALL FROM Customer EXCEPT CustURL, CustEMail
```

The below selects all columns from the *Employee* table where the sales bonus are greater than \$200.00

```
SELECT ALL FROM Employee EXCEPT HireDate, EntryDate WHERE EmpID IN (SELECT EmpID FROM SalesBonus WHERE Bonus > 200.00)
```

1.20.7.7 LIMIT

The LIMIT clause can be used to constrain the number of rows returned by the [SELECT](#) statement. LIMIT takes one or two numeric arguments, which must be integer.

```
SELECT * FROM tablename LIMIT offset, row_count
```

The LIMIT option does not impact how LIMIT is used when it is part of the [WHERE](#) Clause.

With two arguments, the first argument specifies the offset of the first row to return, and the second specifies the maximum number of rows to return. The offset of the initial row is 0 (not 1):

The following retrieves rows 6-15:

```
SELECT * FROM table LIMIT 5,10
```

To retrieve all rows from a certain offset up to the end of the result set, you can use some large number for the second parameter. This statement retrieves all rows from the 96th row to the last:

```
SELECT * FROM table LIMIT 95,99999999
```

With one argument, the value specifies the number of rows to return from the beginning of the result set, which retrieves the first 5 rows:

```
SELECT * FROM table LIMIT 5
```

In other words, LIMIT n is equivalent to LIMIT 0,n.

SELECT ... LIMIT is processed similar to how the [TOP](#) option is processed. These following two commands are now equivalent:

```
SELECT TOP 5 ALL FROM SalesBonus ORDER BY NetAmount DESC
```

```
SELECT ALL FROM SalesBonus ORDER BY NetAmount DESC LIMIT 5
```

Examples:

Because of the 2 value LIMIT option, the following is supported, which has no "TOP" equivalent:

```
SELECT ALL FROM SalesBonus ORDER BY NetAmount DESC LIMIT 5,3
```

SELECT also supports the use of LIMIT with a WHERE clause. The following examples now work:

```
SELECT ALL FROM SalesBonus ORDER BY NetAmount DESC LIMIT 5 WHERE Bonus > 0
```

```
SELECT ALL FROM SalesBonus WHERE Bonus > 0 ORDER BY NetAmount DESC LIMIT 5
```

1.20.7.8 INNER JOIN

This clause is used to retrieve data from two or more tables.

```
SELECT ... FROM lefttblview [corr_name] ...  
      ... INNER JOIN righttblview [corr_name] ...  
      ... ON [lefttblview.corr_name] = [righttblview.corr_name].column2 ...  
      ... [WHERE Clause] ...
```

Options**.column1**

Defines the column on which to link.

.column2

Defines the column on which to link.

corr_name

A correlation name is an alias or nickname for a table. It lets you refer to the same table twice in one command, use a shorter name, and explicitly refer to a column when referring to the same column if that column appears in more than one table. The correlation name must be at least two characters.

FROM lefttblview

Specifies the left table or view.

lefttblview

Explicitly defines the column on which to link the left table name or view.

INNER JOIN righttblview

Specifies the right table or view.

righttblview

Explicitly defines the column on which to link the right table name or view.

WHERE clause

Limits rows of data. See [WHERE](#).

About JOIN

When you perform a SQL JOIN, you specify one column from each table to join on. These two columns contain data that is shared across both tables. You can use multiple joins in the same SQL statement to query data from as many tables as you like.

JOIN Types

Depending on your requirements, you can do an **"INNER"** join or an **"OUTER"** join. The differences are:

- **INNER JOIN:** This will only return rows when there is at least one row in both tables that match the join condition.
- **LEFT OUTER JOIN:** This will return rows that have data in the left table (left of the JOIN keyword), even if there's no matching rows in the right table.
- **RIGHT OUTER JOIN:** This will return rows that have data in the right table (right of the JOIN keyword), even if there's no matching rows in the left table.
- **FULL OUTER JOIN:** This will return all rows, as long as there's matching data in one of the tables.

Nested JOINS

Any of the JOIN types can be mixed in any sequence to create a nested join. The nested joins still require that you specify one column from each table to join on. When nesting joins, it is important to use the correct sequence of parenthesis, along with a correlation for each join.

In the example below, notice the two sets of parenthesis, which all begin after the FROM keyword and end after the linking columns. Also note the "J1" and "J2" correlations specified for each join.

```
SELECT ALL FROM (( TABLE1 t1 +
  INNER JOIN TABLE2 t2 ON t1.FieldT2=t2.FieldT2) J1 +
  INNER JOIN TABLE3 t3 ON t3.FieldT3=j1.FieldT3) J2
```

Examples

INNER JOIN Example:

The following example list an employee's total sales for each day.

```
SELECT t1.empid, t2.netamount, t2.transdate FROM employee t1 +
  INNER JOIN transmaster t2 ON t1.empid = t2.empid +
WHERE empid = 129
```

t1.empid	t2.netamount	t2.TransDate
129	\$3,080.00	07/02/2003
129	\$5,385.00	07/08/2003
129	\$6,160.00	07/11/2003
129	\$5,575.00	08/24/2003
129	\$10,445.00	08/24/2003
129	\$10,175.00	08/25/2003
129	\$2,195.00	08/27/2003

Nested INNER JOIN Example:

The following example lists a specific product, all of the locations where it resides, and the components used within the product.

```
SELECT ProdName, Location, CompID FROM +
  ((Product t1 INNER JOIN ProdLocation t2 ON t1.Model=t2.Model) J1 +
  INNER JOIN CompUsed t3 ON t3.Model=j1.Model) J2 +
WHERE Model = 'CX3000'
```

ProdName	Location	CompID

Standard SVGA Color PC	A-1	X1010
Standard SVGA Color PC	A-1	X2000
Standard SVGA Color PC	A-1	X3000
Standard SVGA Color PC	B-1	X1010
Standard SVGA Color PC	B-1	X2000
Standard SVGA Color PC	B-1	X3000
Standard SVGA Color PC	C-10	X1010
Standard SVGA Color PC	C-10	X2000
Standard SVGA Color PC	C-10	X3000

1.20.7.9 OUTER JOIN

This clause is used to retrieve data from two or more tables.

```

SELECT ... FROM lefttblview [corr_name] [ LEFT
                                RIGHT
                                FULL ] ...

    ... OUTER JOIN righttblview [corr_name] ...

    ... ON [ lefttblview ].column1 = [ righttblview ].column2 ...

    ... [ WHERE clause ] ...

```

Options

.column1

Defines the column on which to link.

.column2

Defines the column on which to link.

corr_name

A correlation name is an alias or nickname for a table. It lets you refer to the same table twice in one command, use a shorter name, and explicitly refer to a column when referring to the same column if that column appears in more than one table. The correlation name must be at least two characters.

FROM lefttblview

Specifies the left table or view.

LEFT

RIGHT

FULL

Specifies the type of outer join.

lefttblview

Explicitly defines the column on which to link the left table name or view.

OUTER JOIN righttblview

Specifies the right table or view.

righttblview

Explicitly defines the column on which to link the right table name or view.

WHERE clause

Limits rows of data. See [WHERE](#).

About JOIN

When you perform a SQL JOIN, you specify one column from each table to join on. These two columns contain data that is shared across both tables. You can use multiple joins in the same SQL statement to query data from as many tables as you like.

JOIN Types

Depending on your requirements, you can do an ["INNER"](#) join or an **"OUTER"** join. The differences are:

- **INNER JOIN:** This will only return rows when there is at least one row in both tables that match the join condition.
- **LEFT OUTER JOIN:** This will return rows that have data in the left table (left of the JOIN keyword), even if there's no matching rows in the right table.
- **RIGHT OUTER JOIN:** This will return rows that have data in the right table (right of the JOIN keyword), even if there's no matching rows in the left table.
- **FULL OUTER JOIN:** This will return all rows, as long as there's matching data in one of the tables.

About OUTER JOIN

When you use an outer join, rows are not required to have matching values. The table order in the FROM clause specifies the left and right table. You can include a WHERE clause and other SELECT clause options such as [GROUP BY](#). The result set is built from the following criteria:

- In all types of outer joins, if the same values for the linking columns are found in each table, R:BASE joins the two rows.
- For a left outer join, R:BASE uses each value unique to the left (first) table and completes it with nulls for the columns of the right (second) table when the linking columns do not match.
- A right outer join uses unique values found in the right (second) table and completes the rows with nulls for columns of the left (first) table when the linking columns do not match.
- A full outer join first joins the linking values, followed by a left and right outer join.

Nested JOINS

Any of the JOIN types can be mixed in any sequence to create a nested join. The nested joins still require that you specify one column from each table to join on. When nesting joins, it is important to use the correct sequence of parenthesis, along with a correlation for each join.

In the example below, notice the two sets of parenthesis, which all begin after the FROM keyword and end after the linking columns. Also note the "J1" and "J2" correlations specified for each join.

```
SELECT ALL FROM (( TABLE1 t1 +
LEFT OUTER JOIN TABLE2 t2 ON t1.FieldT2=t2.FieldT2) J1 +
RIGHT OUTER JOIN TABLE3 t3 ON t3.FieldT3=j1.FieldT3) J2
```

Examples

The following example lists all of the employees and their total sales, including those employees who have not yet completed a sale.

```
SELECT t1.empid, SUM(t2.netamount) FROM employee t1 +
FULL OUTER JOIN transmaster t2 ON t1.empid = t2.empid +
GROUP BY t1.empid
```

t1.empid	SUM(t2.netamount)
102	\$387,060.00
129	\$347,775.00
131	\$472,000.00
133	\$133,140.00
160	\$344,550.00

165	\$29,370.00
166	\$0.00
167	\$3,830.00

Nested INNER JOIN Example:

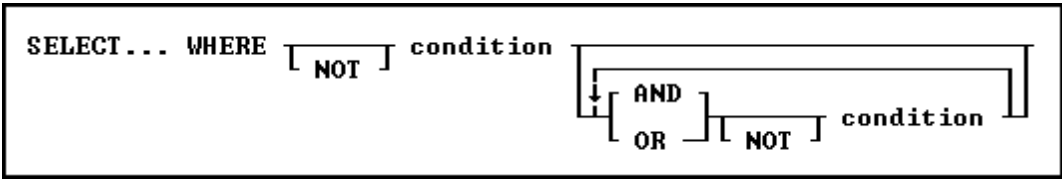
The following example lists a specific product, all of the locations where it resides, and the components used within the product.

```
SELECT ProdName, Location, CompID FROM +
((Product t1 RIGHT OUTER JOIN ProdLocation t2 ON t1.Model=t2.Model) J1 +
RIGHT OUTER JOIN CompUsed t3 ON t3.Model=j1.Model) J2 +
WHERE Model = 'CX3000'
```

ProdName	Location	CompID
-----	-----	-----
Standard SVGA Color PC	A-1	X1010
Standard SVGA Color PC	B-1	X1010
Standard SVGA Color PC	C-10	X1010
Standard SVGA Color PC	A-1	X2000
Standard SVGA Color PC	B-1	X2000
Standard SVGA Color PC	C-10	X2000
Standard SVGA Color PC	A-1	X3000
Standard SVGA Color PC	B-1	X3000
Standard SVGA Color PC	C-10	X3000

1.20.7.10 WHERE

This clause determines which rows of data to include.



Options

AND

Indicates the following condition must be met along with the preceding condition.

condition

Identifies requirements to be in the WHERE syntax.

NOT

Reverses the meaning of a connecting operator. AND NOT, for example, indicates that the first condition must be met and the following condition must not be met.

OR

Indicates the following condition can be met instead of the preceding condition.

About the WHERE Clause

In most commands, a WHERE clause follows the syntax diagram above. The [JOIN](#) command does not use that syntax diagram because it uses only comparison operators in a WHERE clause to compare two columns.

The two main elements in any WHERE clause are conditions and connecting operators.

R:BASE supports "COUNT = LAST" in two different methods. If the entire WHERE clause is "WHERE COUNT = LAST" then R:BASE works like it always has, to quickly fetch the last row of the table. The added functionality is to have other conditions in the WHERE clause, and you want the last row of whatever qualifies.

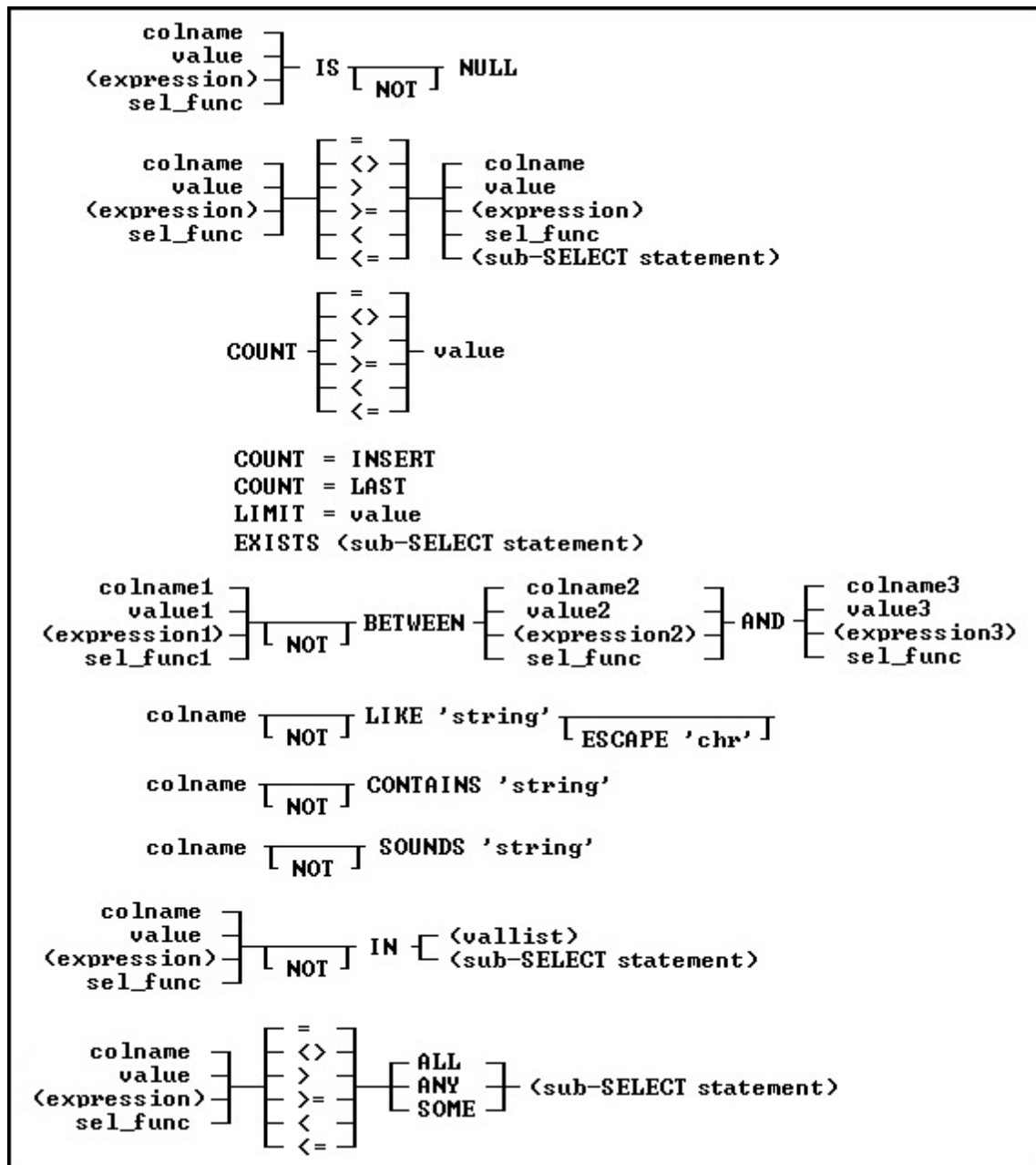
To make it work this way specify the other conditions and then add "AND COUNT = LAST".

Here is an example:

```
SELECT * FROM Customer WHERE CustID > 100 AND COUNT = LAST
```

WHERE Clause Conditions

The following syntax diagram and table show the basic formats for WHERE clause conditions, which can be used alone or together.



Basic WHERE Clause Conditions

Condition Syntax	Description
<i>colname op</i> DEFAULT	True if a column value compares correctly with the DEFAULT value for the column. <i>Op</i> can be =, <>, >=, >, <=, or <.
<i>colname</i> = USER	True if a column value equals the current user identifier.
<i>item1</i> IS NULL	True if <i>item1</i> has a null value. <i>Item1</i> can be a column name, value, or expression. A null value cannot be used in a comparison with an operator.
<i>item1 op item2</i>	True if the relationship between two items is true as defined by an operator. <i>Item1</i> can be a column name, value, or expression; <i>item2</i> can be a column name, value, expression, or sub-SELECT statement.
COUNT=INSERT	Refers to the last row inserted in a table by the current user, even if it has been modified by another user. The COUNT=INSERT condition can be used with a single-table view, but not with a multi-table view. If there is not a newly inserted row in the table, then COUNT=INSERT performs the same action as COUNT=LAST, and fetches the current end row of the table.
COUNT=LAST	Refers to the last row in a table. The COUNT=LAST condition can be used with a single-table view, but not with a multi-table view.
COUNT <i>op</i> <i>value</i>	Refers to a number of rows defined by <i>op</i> and <i>value</i> .
LIMIT= <i>value</i>	Specifies a number of rows affected by a command. A LIMIT condition should be the last condition in a WHERE clause.
EXISTS (sub-SELECT statement)	True if sub-SELECT statement returns one or more rows.
<i>item1</i> BETWEEN <i>item2</i> AND <i>item3</i>	True if the value of <i>item1</i> is greater than or equal to the value of <i>item2</i> , and if the value of <i>item1</i> is less than or equal to the value of <i>item3</i> .
<i>colname</i> LIKE ' <i>string</i> '	True if a column value equals the text string. With LIKE, a string can also be a DATE, TIME, or DATETIME value. The text string can contain R:BASE wildcard characters.
<i>colname</i> LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	True if a column value equals a text string. If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>colname</i> CONTAINS ' <i>string</i> '	True if a column value contains the text string. The text string can contain R:BASE wildcard characters.
<i>colname</i> SOUNDS ' <i>string</i> '	True if the soundex value of a column matches the soundex value of the text string.
<i>item1</i> IN (<i>vallist</i>)	True if <i>item1</i> is in the value list.
<i>item1</i> IN (sub-SELECT statement)	True if <i>item1</i> is in the rows selected by a sub-SELECT.
<i>item1 op</i> ALL (sub-SELECT statement)	True if the relationship between <i>item1</i> and every row returned by a sub-SELECT statement matches an operator.
<i>item1 op</i> ANY(sub-SELECT statement)	True if the relationship between <i>item1</i> and at least one value returned by a sub-SELECT statement matches an operator.
<i>item1 op</i> SOME (sub-SELECT statement)	ANY and SOME are equivalent.

Notes:

- Placing NOT before most text operators (such as NULL or BETWEEN) reverses their meaning.
- When a SELECT statement is part of a WHERE clause, it is called a [sub-SELECT](#) clause. A sub-SELECT clause can contain only one column name (not a column list or *), expression, or function. The [INTO](#) and [ORDER BY](#) clauses in a sub-SELECT are ignored.

You can only use the current wildcard characters to compare a column to a text value when using the LIKE comparison. The default wildcard characters are the percent sign (%), which is used for one or more characters, and the underscore (_), which is used for a single character.

If you compare a column with a value, you can either enter the value or specify a global variable. If you specify a variable, R:BASE compares the column with the current value of the variable.

To significantly reduce processing time for a WHERE clause, use INDEX processing. To use indexes, the following conditions must be met:

- A condition in the WHERE clause compares an indexed column.
- If the WHERE clause contains more than one condition, R:BASE selects the condition that places the greatest restriction on the WHERE clause.
- Conditions are not joined by the OR operator.
- The comparison value is not an expression.

Connecting Operators

When you use more than one condition in a WHERE clause, the conditions are connected using the connecting operators AND, OR, AND NOT, and OR NOT.

The connecting operator AND requires that both conditions it separates must be satisfied. The connecting operator OR requires that either condition it separates must be satisfied.

The connecting operator AND NOT requires that the preceding condition must be satisfied, and the following condition must not be satisfied. The connecting operator OR NOT requires that either the preceding condition must be satisfied, or any condition except the following condition must be satisfied.

In WHERE clauses with multiple conditions, conditions that are connected by AND or AND NOT are evaluated before those connected by OR or OR NOT. However, you can control the order in which conditions are evaluated by either placing parentheses around conditions or using the [SET AND](#) command. If you set AND off, conditions are always evaluated from left to right.

Examples

The following WHERE clause chooses sales amounts that are less than the value of a variable containing the daily average.

```
... WHERE amount < .dailyave
```

The following WHERE clause specifies the seventh row.

```
... WHERE COUNT = 7
```

The following WHERE clause specifies each row from the *employeetable* that contains both the first name *June* and the last name *Wilson*.

```
SELECT * FROM employee WHERE empfname = 'june' AND emplname = 'wilson'
```

The following WHERE clause selects dates in the *actdate* column that are greater than dates in the *begdate* column or are less than dates in the *enddate* column.

```
... WHERE actdate BETWEEN begdate AND enddate
```

The next three WHERE clauses use the following data:

```
empfname emplname
-----
Mary Jones
John Smith
Agnes Smith
John Brown
```

In both of the following clauses, R:BASE first evaluates the conditions connected by AND, selecting John Smith. Then R:BASE adds any Marys to the list because the connecting operator is OR. The final result includes John Smith and Mary Jones.

```
...WHERE empfname = 'Mary' OR empfname = 'John' +
AND emplname = 'Smith'
```

```
...WHERE empfname = 'Mary' OR (empfname = 'John' +
AND emplname = 'Smith')
```

By moving the parentheses around the conditions connected by OR, you can select only John Smith. In the following WHERE clause, the first name can be either Mary or John, but the last name must be Smith.

```
...WHERE (empfname = 'Mary' OR empfname = 'John') AND +
emplname = 'Smith'
```

The following example illustrates a sub-SELECT in a WHERE clause. Assume you wanted a list of all sales representatives that had transactions greater than \$100,000, and the information for such a list was contained in two tables, *employee* and *transmaster*. The relevant columns in these tables are:

employee	transmaster
empid	emplname empid netamount
102	Wilson 133 \$32,400.00
129	Hernandez 160 \$9,500.00
133	Coffin 129 \$6,400.00
165	Williams 102 \$176,000.00
166	Chou 160 \$194,750.00
167	Watson 129 \$34,125.00
160	Smith 131 \$152,250.00
131	Simpson 102 \$87,500.00
102	\$22,500.00
102	\$40,500.00
131	\$108,750.00
131	\$80,500.00
129	\$56,250.00
102	\$57,500.00
160	\$140,300.00
129	\$95,500.00
129	\$155,500.00
133	\$88,000.00
131	\$130,500.00
102	\$3,060.00
165	\$3,060.00
167	\$3,830.00
133	\$12,740.00
165	\$26,310.00

To display a list of employees in the *transmaster* table with a transaction larger than \$100,000, enter the following command:

```
SELECT empid, emplname FROM employee WHERE empid IN +
(SELECT empid FROM transmaster WHERE netamount > 100000)
```

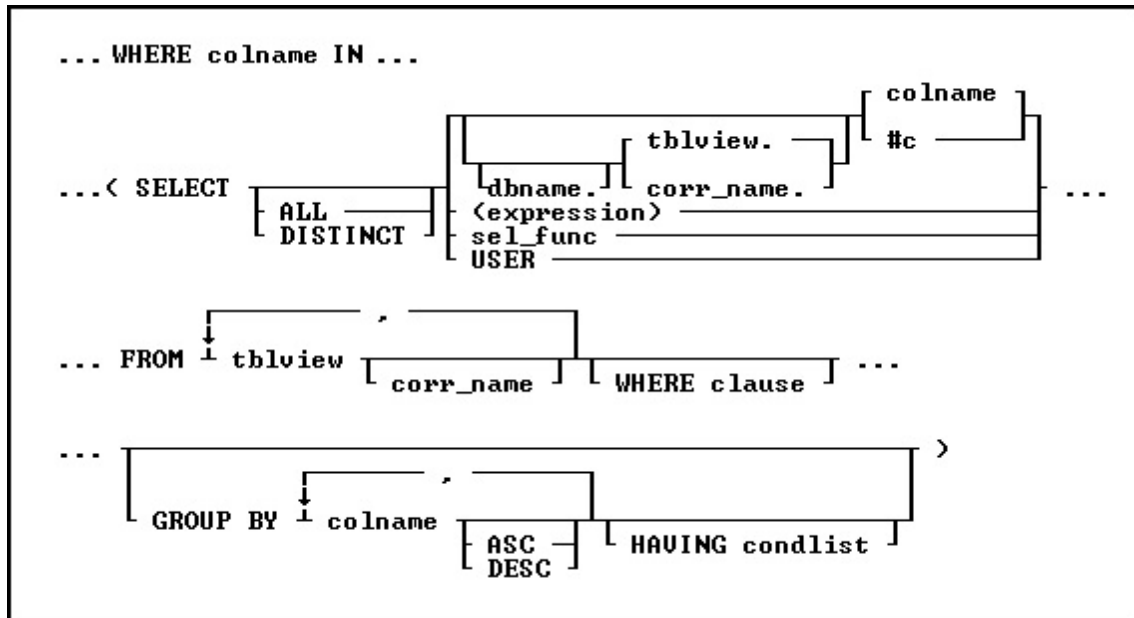
R:BASE displays the following list:

empid	emplname
102	Wilson
129	Hernandez
131	Simpson
160	Smith

Note: You can use a [sub-SELECT](#) in any command that allows a full WHERE clause.

1.20.7.11 Sub-SELECT

This clause works in conjunction with the [WHERE](#) clause to determine which rows of data to include.



Options

For a description of the options, see [SELECT](#).

About the Sub-SELECT Clause

A sub-SELECT command, which is a SELECT command nested within another command, always appears in a [WHERE](#) clause, whether the sub-SELECT command is nested in the WHERE clause of a SELECT command or in another command such as [EDIT USING form](#). R:BASE processes the clauses in a sub-SELECT in the same order as in a SELECT command. The order (ascending/descending) is ignored within a Sub-SELECT. All control of ordering should be in the outermost SELECT.

Examples

The following example selects customers that have purchased items in the month of March within the ConComp sample database:

```
SELECT company FROM customer WHERE custid +
IN (SELECT custid FROM transmaster +
WHERE (IMON(transdate)=3))
```

```
company
PC Distribution Inc.
Computer Distributors Inc.
Industrial Concepts Inc.
PC Consultation and Design
```

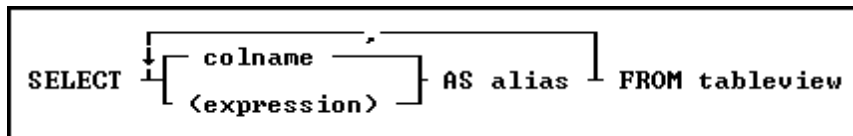
The following example selects and groups employee and customer IDs for purchases sold by the employee Ernest Hernandez, within the ConComp sample database

```
SELECT empid, custid FROM transmaster WHERE empid IN +
(SELECT empid FROM Employee WHERE EmpLName = 'Hernandez') +
GROUP BY empid, custid
```

empid	custid
129	105
129	110
129	113
129	117
129	124

1.20.7.12 AS

This clause dynamically renames columns in a [SELECT](#) clause.



Options

'
Indicates that this part of the command is repeatable.

colname

Specifies a column name. The column name is limited to 128 characters.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

AS alias

Determines the alias of the column. This may be used to refer to the column in other locations.

FROM tableview

Specifies the table or view to draw information from.

About the SELECT AS command

The SELECT AS command functions exactly like any other SELECT command and will accept all other SELECT options. The one noticeable exception is that this allows you to give a column an alias. This can be most useful when used in conjunction with the [CREATE VIEW](#) command.

Examples

The following 3 examples are based on the *ConComp* database.

The following command selects the EmpID and the EmpName columns from the Employee table and renames them to "EmployeeID" and "Name".

```
SELECT EmpID AS EmployeeID, EmpName AS Name FROM Employee
```

The following command creates a VIEW using the SELECT AS notation. This view contains a column for Employee ID, Employee Name (which is a single column based on the EmpFName and EmpLName columns) from the Employee table and the Transdate and NetAmount columns from the Transmaster table. The immediately following command browses the Employee Name, Transdate and Netamount column. For more on Views please see [CREATE VIEW](#).

```
CREATE VIEW EmpAmount AS SELECT T1.EmpID, +
    (T1.EmpFName + ' ' + T1.EmpLName) AS EmpName, +
    T2.TransDate, T2.NetAmount FROM Employee T1, +
```

```
TransMaster T2 WHERE T1.EmpID = T2.EmpID
```

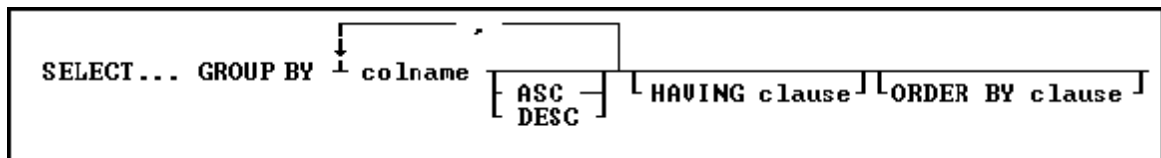
```
BROWSE EmpName,TransDate,NetAmount FROM EmpAmount
```

This final example uses IDQuotes to create a column name with spaces in it and then uses the [SELECT HTML](#) option to turn that into an HTML table with the column name. The [OUTPUT](#) commands redirect output to a file called "Emp.HTM" and then back to the screen. We do NOT recommend using this method to create VIEWS or TABLES with names that contain spaces as this could lead to Database Corruption.

```
OUTPUT EMP.HTM
SELECT EmpID as `Employee ID` FROM Employee HTML
OUTPUT SCREEN
```

1.20.7.13 GROUP BY

This clause determines which rows of data to include.



Options

,

Indicates that this part of the command is repeatable.

ASC DESC

Specifies whether to sort a column in ascending or descending order.

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (tblname.colname).

GROUP BY

Returns a group of rows as a summary resulting in only unique rows. This option is generally used with [SELECT Functions](#).

HAVING clause

Limits the rows affected by the GROUP BY clause.

ORDER BY clause

Sorts rows of data.

About the GROUP BY command

This optional clause groups rows according to the values in one or more columns and sorts the results. GROUP BY consolidates the information from several rows into one row. This results in a table with one row for each value in the named column or columns and one or more values per column.

The columns listed in the GROUP BY clause are related to those listed in the command clause. Any column named in the GROUP BY clause can also be named in the command clause, but any column not named in the GROUP BY clause can be used only in the command clause if the column is used in a SELECT function.

Examples

The [SELECT](#) command clause can contain the columns named in the GROUP BY clause, and SELECT functions that refer only to columns not named in the GROUP BY clause. Because the GROUP BY clause processes information resulting from a [WHERE](#) clause, you can add a GROUP BY clause to see the sales each employee has made:

```
SELECT empid FROM transmaster WHERE netamount < $100,000 +
GROUP BY empid
```

The following intermediate result table contains columns not named in the command clause because the command clause has not been processed yet (not all the columns fit in the display, however). The first part of the processing is to group the rows by *empid*. Because seven different employees are included, the intermediate result table includes seven rows.

Intermediate Result Table-GROUP BY empid

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
4975, 4980, 5000, 5060, 5045	101, 101, 101, 101, 100	102	\$87,500, \$22,500, \$40,500, \$57,500, \$3,060
4790, 4865, 5050, 5070	104, 102, 104, 104	129	\$6,400, \$34,125, \$56,250, \$95,500
5015	103	131	\$80,500
4760, 5080, 5048	100, 100, 103	133	\$32,400, \$88,000, \$12,740
4780	105	160	\$9,500
5046, 5049	101, 102	165	\$3,060, \$26,310
5047	102	167	\$3,830

You can include more than one column in a GROUP BY clause. If you group the rows in the above example by *custid* as well as *empid*, the command looks like this:

```
SELECT empid, custid FROM transmaster +
WHERE netamount < $100,000 GROUP BY empid, custid
```

In the following table, rows are now grouped by both *empid* and *custid*, resulting in eleven groups.

Intermediate Result Table-GROUP BY empid and custid

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
5045	100	102	\$3,060
4975, 4980, 5000, 5060	101	102	\$87,500, \$22,500, \$40,500, \$57,500
4865	102	129	\$34,125
4790, 5050, 5070	104	129	\$64,000, \$56,250, \$95,500
5015	103	131	\$80,500
4760, 5080	100	133	\$32,400, \$88,000
5048	103	133	\$12,740
4780	105	160	\$9,500
5046	101	165	\$3,060
5049	102	165	\$26,310
5047	102	167	\$3,830

If one or more of the columns named in the GROUP BY clause contain null values, R:BASE forms a separate group for null values. Review the result of this [SELECT](#) command for the *employee* table:

```
SELECT empid, emplname, hiredate, emptitle FROM employee
```

<u>empid</u>	<u>emplname</u>	<u>hiredate</u>	<u>emptitle</u>
102	Wilson	03/18/90	Manager
129	Hernandez	08/28/91	Manager
131	Smith	04/14/92	-0-
133	Coffin	11/26/93	Representative
160	Simpson	01/09/94	-0-
165	Williams	07/05/92	Representative
167	Watson	07/10/92	Representative
166	Chou	07/10/93	Sales Clerk

If you group these rows by the *emptitle* column, which contains null values, you get the following intermediate result table:

Intermediate Result Table-GROUP BY emptitle

<u>empid</u>	<u>emplname</u>	<u>hiredate</u>	<u>emptitle</u>
102, 129	Wilson, Hernandez	03/18/90, 08/28/91	Manager
133, 165, 167	Coffin, Williams, Watson	11/26/93, 07/05/92, 07/10/92	Representative
166	Chou	07/10/93	Sales Clerk
131, 160	Smith, Simpson	04/14/94, 01/09/94	-0-

1.20.7.14 HAVING

This clause determines which rows of data to include based on the results of a prior [GROUP BY](#) clause.

```
SELECT ... HAVING [ NOT ] condition
                        [ AND ]
                        [ OR ] [ NOT ] condition
```

Options

AND OR

AND indicates two conditions must both be true.

OR indicates either condition must be true.

condition

Specifies a combination of one or more expressions and/or operations that would evaluate to either true or false. See "HAVING Conditions" below.

NOT

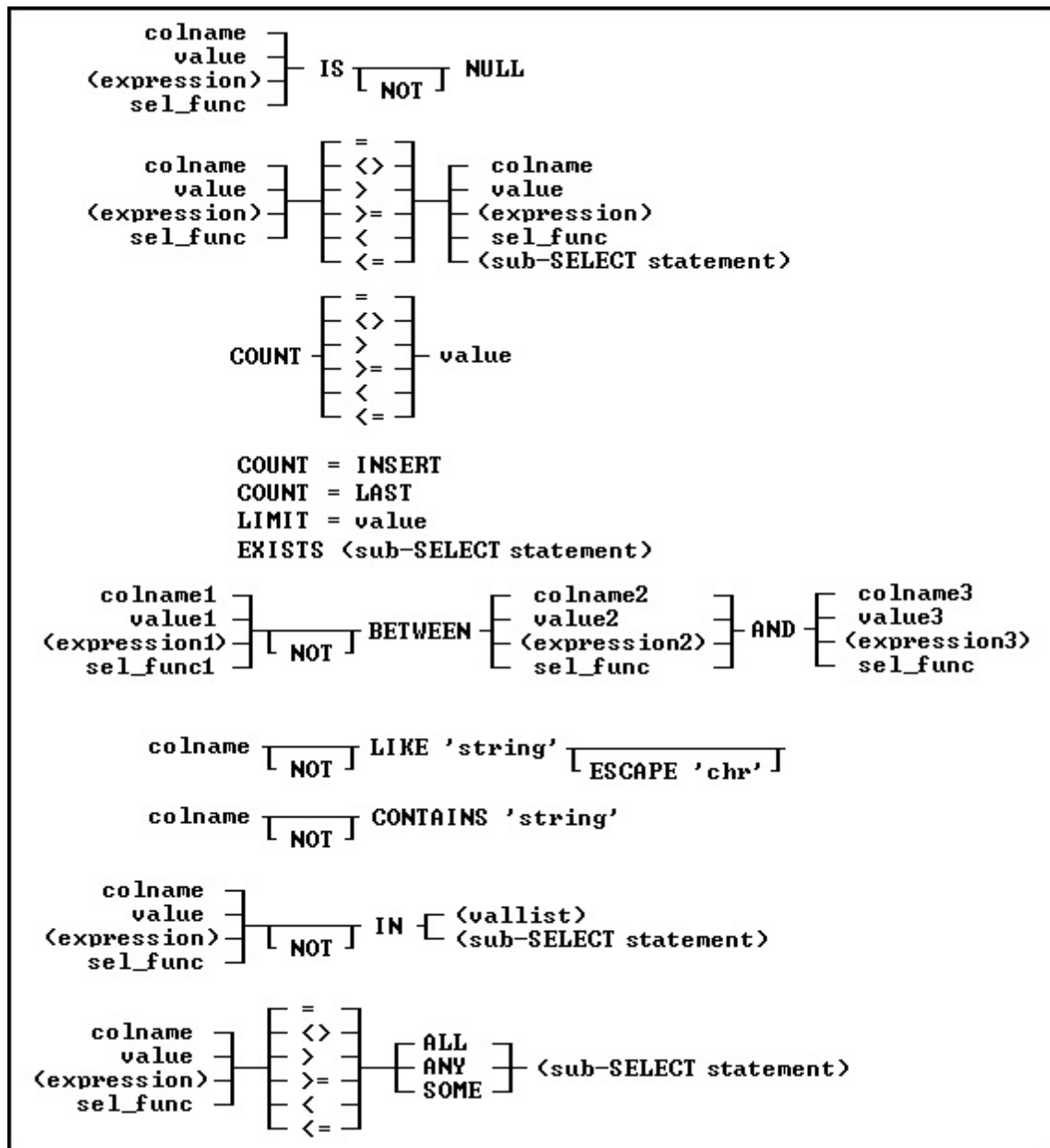
Reverses the meaning of an operator or indicates that a condition is not true.

About the HAVING command

The optional HAVING clause selects rows that meet one or more conditions from among the results of the [GROUP BY](#) clause. HAVING works the same as a WHERE clause with the following exceptions:

- A WHERE clause modifies the intermediate results of a FROM clause; a HAVING clause modifies the intermediate results of a GROUP BY clause.
- A HAVING clause can include [SELECT Functions](#).

HAVING Conditions:



Examples

To display sales information for only those employees who have made more than one sale to the same customer, add a HAVING clause such as the following to one of the examples shown in [GROUP BY](#). When used in a HAVING clause, [SELECT functions](#) compute results based on the values grouped in the specified column. In this HAVING clause, COUNT returns the number of values grouped in the *transid* column.

```
SELECT empid, custid FROM transmaster +
WHERE netamount < $100,000 +
GROUP BY empid, custid HAVING COUNT(transid) > 1
```

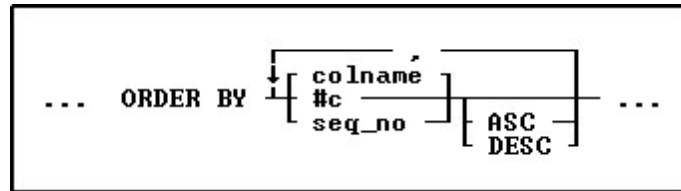
Intermediate Result Table-HAVING COUNT(transid) > 1

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
4975, 4980, 5000, 5060	101	102	\$87,500, \$22,500, \$40,500, \$57,500

4790, 5050, 5070	104	129	\$6,400, \$56,250, \$95,500
5080	100	133	\$32,400, \$88,000

1.20.7.15 ORDER BY

This clause specifies how to sort the result of the [SELECT](#) command.



Options

' Indicates that this part of the command is repeatable.

ASC DESC

Specifies whether to sort a column in ascending or descending order.

#c

Takes the place of a column name and refers to the column numbers displayed with the [LIST TABLE](#) command.

colname

Sorts by any column name or combination of column names.

seq_no

Refers to the items listed in the [SELECT](#) command that is using the ORDER BY command, ordered from left to right. An item can be a column name, expression, or [SELECT function](#).

About the ORDER BY Command

The syntax for the ORDER BY clause is the same for all commands. ORDER BY must refer to only one table or view.

You can significantly reduce the time R:BASE takes to process an ORDER BY clause when the column or columns listed in the ORDER BY clause are included in an index with the same column sort order as that specified in the ORDER BY clause.

Using the SET SORT Command

The ORDER BY command uses the R:BASE automatic sort optimizer. If you are sorting extremely large tables, and if your disk space is limited, the automatic sort optimizer might be unable to sort the data. Instead, use the [SET SORT ON](#) condition because it uses the least disk space necessary to sort data; however, the SET SORT ON condition is slower than the automatic sort.

Examples

The following command displays data from the *custid*, *company*, and *custcity* columns from the *customertable*.

```
SELECT custid, company, custcity FROM customer
```

The ORDER BY clause in the command below arranges the *custid* values in descending order.

```
SELECT custid, company, custcity FROM customer +
ORDER BY custid DESC
```

You can substitute a column's sequence number for a column named in the ORDER BY clause. You must use a sequence number when referring to an expression, function, constant, or when a UNION operator is used. The following command is equivalent to the command example above.

```
SELECT custid, company, custcity FROM customer ORDER +
BY 1 DESC
```

1.20.7.16 UNION

You can use this operator to combine the results of two or more [SELECT](#) statements.

```
SELECT clause FROM clause... UNION [ ALL ] SELECT clause FROM clause
```

About the UNION SELECT command

This optional operator combines the results of two SELECT commands or clauses, displaying the results of the second SELECT command below those of the first. By default, UNION deletes duplicate rows. Include the optional keyword ALL to include duplicate rows in the final result. You cannot combine [sub-SELECT](#) commands using UNION.

The UNION operator requires the following three conditions:

- The SELECT statements must specify an equal number of columns.
- Columns that are being combined must have the same data type.
- Only the last SELECT statement can contain an ORDER BY clause.

To use a UNION SELECT to combine two text columns of different lengths, put parentheses around the column names to have R:BASE interpret the values as expressions. The same technique can be used to combine NOTE and TEXT data types values.

Examples

Example 01:

The following example lists all employees and the sales transactions for each, including those employees who have not yet completed a sale.

```
SELECT employee.empid, transid +
FROM employee, transmaster +
WHERE employee.empid = transmaster.empid +
UNION SELECT empid, 0 +
FROM employee +
WHERE employee.empid NOT IN +
(SELECT empid FROM transmaster)
```

The first SELECT displays the *empid* column from the *employee* table and *transid* from the *transmaster* table, linking the tables by the common column, *empid*. In short, the first SELECT displays all employees who have made a sale.

The second SELECT command selects the *empid* column from *employee*, including rows only for those employees who are not listed in the *transmaster* table. Because the results of the second SELECT are appended to those of the first (by the UNION operator), those employees who have not yet made a sale are shown at the bottom of the results with a zero in the *transid* column. The final results look like this:

```
empid  transid
-----
102    4795
102    4975
102    4980
```

```

102    5000
102    5045
102    5060
129    4790
129    4865
129    5050
129    5070
129    5075
131    4970
131    5010
131    5015
131    5085
133    4760
133    5048
133    5080
160    4780
160    4800
160    5065
165    5046
165    5049
166    0

```

Example 02:

The following example combines two tables where the 'X' in the first part of the SELECT is treated as an expression, which means that it is created as a NOTE data type, because that is the data type which supports variable length TEXT values as a result of an expression. The DYCD_AcctNum column is an actual column of type TEXT and length 1. As columns that are being combined must have the same data type, putting parentheses around the DYCD_AcctNum column makes it an expression as well.

```

SELECT Major, GLDesc=50, 'X' AS Status +
  FROM GLAcctNum +
UNION SELECT Major, GLDesc=50, (DYCD_AcctNum) AS Status +
  FROM HHSAcctNum

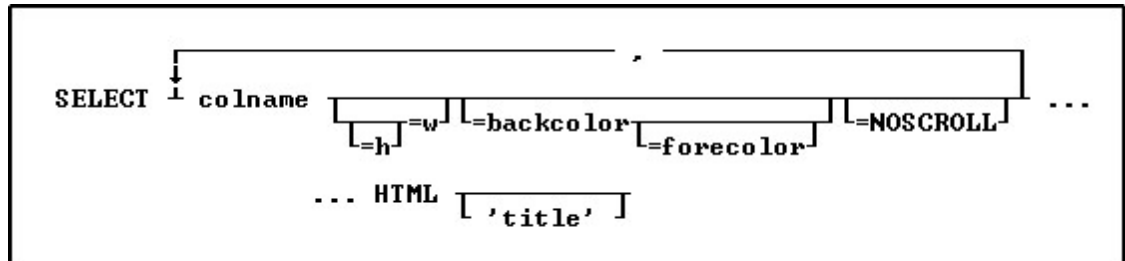
```

Major	GLDesc	Status
4	PAYROLL CASH ACCOUNT	X
5	CLAIMS CASH ACCOUNT	X
42	PETTY CASH FUND ACCOUNT	X
53	ACCOUNT RECEIVABLE	X
53	OTHER RECEIVABLE	X
53	TAX REFUND DUE	X
100	TOTAL ASSETS	X
200	TAX PAYABLE PASS THROUGH	X
201	FEDERAL TAX PAYABLE	X
206	LOAN - MISC. PAYROLL DEDUCTIONS	X
206	MEDICAL - MISC. PAYROLL DEDUCTIONS	X
206	OTHER -MISC. PAYROLL DEDUCTIONS	X
235	LOANS PAYABLE	X
1100	GROSS SALARIES AND WAGES	X
2410	SAFETY AND HEALTH	X
2420	STAFF TRAINING	X
2430	ACCOUNTING COSTS	X
2440	LEGAL COSTS	X
3200	EQUIPMENT PURCHASE	X
3710	OTHER COSTS	X
3739	OTHER OPERATING COSTS	X

1.20.7.17 HTML

This clause sends the [SELECT](#) command output into HTML format.

The rows and columns that each clause selects produce an intermediate result table that exists only in temporary memory. One after another, the clauses restrict the rows or columns included in the intermediate result table. After R:BASE has processed all the clauses and any [UNION](#) operators, the intermediate result table becomes the final result table.



Options

=backcolor

Specifies the background color for the data in the column.

colname

Specifies a column name. The column name is limited to 128 characters.

=forecolor

Specifies the text color for the data in the column. If you only specify one color, it is used as the background color.

=h =w

Specifies height and width parameters. To assign the same justification for both data and the header, the keywords LEFT, RIGHT, and CENTER can be used. For width, the range is from 0-255 pixels. 0 is no width specification. The height is the addition of the following values, one value from each item:

data justification:	0=default, 1=left, 2=center, 3=right
heading justification:	0=default, 4=left, 8=center, 12=right
vertical alignment:	0=default, 16=top, 32=middle, 48=bottom
HTML format flag:	0=no, 64=yes

HTML

Converts the data to HTML code.

=NOSCROLL

Specifies a NOWRAP tag is added to prevent word-wrapping in the table cell.

'title'

Specifies the text that appears in the caption at the top of the Web Browser window. Adding 'title' creates the beginning and ending table tag as well as putting the text you enter in the caption. Without 'title', a partial HTML file with the selected data in table row format is generated.

About the SELECT...HTML Command

The SELECT...HTML command is a modification of the [SELECT](#) command to output data in HTML format. If a title is specified, a full HTML file will be generated. Otherwise, a partial HTML file with the selected data in table row format will be generated.

Available Colors:

- Black
- Blue
- Green

- Cyan
- Red
- Magenta
- Brown
- Gray
- Light Black
- Light Blue
- Light Green
- Light Cyan
- Light Red
- Light Magenta
- Yellow
- White

Examples:

Example 01:

--Selects "Print Option" data into HTML output with the beginning and ending table tag

```
OUTPUT PrintOptions.htm
SELECT Option=150=CENTER=BLACK=BROWN, DisplayName=200=CENTER=BLACK=BROWN FROM
PrintOptions HTML 'Print Options'
OUTPUT SCREEN
```

Example 02:

--Selects "Customer" data into HTML output using alias names and with just row data. The CustID column is left justified, while the Company column lists the heading as center justified and the data left justified.

```
OUTPUT Customers.htm
WRITE '<HTML>'
WRITE '<TITLE>Customers</TITLE>'
WRITE '<TABLE BORDER=1>'
SELECT CustID=CENTER=100=BLACK=GRAY AS `Customer ID`, Company=9=0=BLACK=GRAY AS
`Company Name` FROM Customer HTML
WRITE '</TABLE>'
WRITE '</HTML>'
OUTPUT SCREEN
```

1.20.8 SET

Use SET to change the current status of R:BASE special characters and operating conditions for any SET category. The SET operating conditions are used to set the database environment while you build and run commands from the R> Prompt or command files and applications.

About the SET Command

To change the setting from the R> Prompt for a special character, enter:

```
SET character_name = value
```

Do not use spaces on either side of the equal sign (=). For example:

```
SET QUOTES='
```

To change the setting from the R> Prompt for an operating condition, enter:

```
SET keyword setting
```

For example:

```
SET BELL ON
```

To remap keys on the keyboard from the R> Prompt, enter:

```
SET KEYMAP keyname = remapped keys
```

For example, if you want to remap [SHIFT] [F6] to [F2], enter:

```
SET KEYMAP [SHIFT] [F6] = [F2]
```

To reset a key to its original default value from the R> Prompt, enter:

```
SET KEYMAP keyname OFF
```

Saving Settings

Settings can be changed at the R> Prompt for the current session. They will revert to the default upon exiting R:BASE. Users can save settings in the R:BASE configuration file. Some settings are stored within the database itself and only the database owner can save changes to the settings that are stored in the database. If you are not the database owner, you can only change the settings that are stored in the database for the current database session.

The R:BASE/Oterro database provides the following categories of SET Keywords:

- Data Integrity
- Display Control
- Environment
- Format
- Programming
- Special Characters
- Transaction Processing and Multi-User
- Database Specific

1.20.8.1 AND

Operating Condition

Syntax: SET AND ON/OFF

Default: ON

SET AND gives the connecting operator AND precedence over OR in [WHERE](#), [IF](#), and [WHILE](#) conditions. When on, R:BASE processes conditions in the following order: all AND NOT followed by all AND OR, and all AND before all OR conditions.

SET AND OFF directs R:BASE to process conditions from left to right. The WHERE clause A AND B OR C AND D is evaluated as (((A AND B) OR C) AND D). SET AND ON directs R:BASE to give the operator AND precedence over OR. The WHERE clause A AND B OR C AND D is evaluated as ((A AND B) OR (C AND D)), which is a different result set.

1.20.8.2 ANSI

Operating Condition

Syntax: SET ANSI ON/OFF

Default: ON

Set ANSI to OFF to allow R:BASE to accept the reserved words.

1.20.8.3 ANSIOUTPUT

Operating Condition

Syntax: SET ANSIOUTPUT ON/OFF

Default: OFF

ANSIOUTPUT specifies ANSI is the default encoding method for the [OUTPUT](#) command. When ON, the ANSI modifier does not need specified for OUTPUT commands. The output file will be created as an ANSI file when it makes sense. If you specified the UTF8 modifier on the OUTPUT command, it will take precedence over the ANSIOUTPUT setting.

1.20.8.4 AUTOCOMMIT

Operating Condition

Syntax: SET AUTOCOMMIT ON/OFF

Default: OFF

Mode: Transaction Processing

SET AUTOCOMMIT toggles AUTOCOMMIT processing on and off. When transaction processing and AUTOCOMMIT are on, each command that is executed successfully is immediately made permanent and visible to network users. If transaction processing is on and AUTOCOMMIT is off, you must enter a [COMMIT](#) command to make changes permanent. Also, leaving the database causes R:BASE to issue the COMMIT command.

When transaction processing is on and AUTOCOMMIT is off, you can enter a series of commands (a transaction) that change data or the database structure, then enter either a COMMIT or [ROLLBACK](#) command. COMMIT makes permanent all changes executed by commands in the transaction. ROLLBACK deletes all the changes, restoring the database to its state before the transaction began.

If you have started a transaction when you set AUTOCOMMIT on, R:BASE commits the transaction and turns AUTOCOMMIT on. You cannot open a [cursor](#) while AUTOCOMMIT is set on, and you cannot set AUTOCOMMIT on while a cursor is open.

AUTOCOMMIT and [MDI](#) cannot be set to on at the same time. If MDI forms and browse screens are to be used, then autocommit mode cannot be running. If autocommit mode is required, then forms and browse screens cannot be modeless.

AUTOCOMMIT can affect system performance. You can increase performance by setting AUTOCOMMIT to on when you do not need to enter commands in groups.

1.20.8.5 AUTOCONVERT

Operating Condition

Syntax: SET AUTOCONVERT ON/OFF

Default: OFF

Set AUTOCONVERT on to automatically convert R:BASE databases created in versions prior to 6.0 to the current R:BASE release. The user is not given the option to halt the conversion process.

1.20.8.6 AUTODROP

Operating Condition

Syntax: SET AUTODROP ON/OFF

Default: OFF

Controls the feature for a combo-box in a form to automatically drop-down the list when it gets focus.

When AUTODROP is SET to ON, this new setting controls the feature for a combo-box in a form to automatically drop-down the list when it gets focus. The editable or non-editable automatic drop-down list of combo box will allow the user to place the value into a column or variable.

This new setting can be saved in your R:BASE configuration file or in your individual application startup file.

Supported environments:

- Configuration File
- Command File
- EEPs

Notes:

- [SHOW](#) AUTODROP will display the current setting of AUTODROP.
- (CVAL('AUTODROP')) will return the current setting of AUTODROP.

1.20.8.7 AUTORECOVER

Operating Condition

Syntax: SET AUTORECOVER ON/OFF

Default: OFF

Mode: Transaction Processing

If AUTORECOVER is set on, errors that can occur during transaction processing when the program is interrupted, for example from a network or power failure, are automatically corrected.

1.20.8.8 AUTOROWVER

Operating Condition

Syntax: SET AUTOROWVER ON/OFF

Default: OFF

AUTOROWVER is used for [Oterro](#) compatibility only. If AUTOROWVER is set on, every [CREATE TABLE](#) or [ALTER TABLE](#) command will add the SYS_ROWVER column if it does not already exist. The SYS_ROWVER column is not compatible with R:BASE 6.0 and lower databases.

1.20.8.9 AUTOSKIP

Operating Condition

Syntax: SET AUTOSKIP ON/OFF

Default: OFF

Set AUTOSKIP on to move the cursor automatically to the next data-entry field in a form after filling the entire field. Specify off to press [Tab] after each entry. R:BASE stores the setting with the database.

1.20.8.10 AUTOSYNC

Operating Condition

Syntax: SET AUTOSYNC ON/OFF

Default: OFF

If AUTOSYNC is set on, connecting to a database will automatically synchronize the database files if necessary. If AUTOSYNC is set off and an error occurs during the connect because the files are out of sync, the database is not connected.

See also:

[RBSYNC](#)

1.20.8.11 AUTOUPGRADE

Operating Condition

Syntax: SET AUTOUPGRADE ON/OFF

Default: OFF

AUTOUPGRADE converts R:BASE 6.0 databases to the current R:BASE release and adds the new system tables for handling stored procedures and triggers.

1.20.8.12 BELL

Operating Condition

Syntax: SET BELL ON/OFF

Default: ON

Set BELL on to sound the bell when an error occurs. Specify off to suppress the bell. R:BASE stores the setting with the database.

1.20.8.13 BLANK

Special Character

Sets the character for spaces. The BLANK character is used to separate words in a command string.

Syntax: SET BLANK=NULL

Syntax: SET BLANK=char
(Use NULL to disable the special character.)

Default: (space)

1.20.8.14 BOOLEAN

Operating Condition

Syntax: SET BOOLEAN ON/OFF

Default: OFF

Set BOOLEAN to ON will specify that constants (e.g. TRUE, FALSE) in expressions will be treated as type BOOLEAN values.

1.20.8.15 CAPTION

Operating Condition

Syntax: SET CAPTION 'window title'

SET CAPTION specifies a title for an application or a command file. This title appears in the title bar when you run the application or command file.

1.20.8.16 CASE

Operating Condition

Syntax: SET CASE ON/OFF

Default: OFF

SET CASE sets the uppercase or lowercase distinction when a comparison is used with [WHERE](#) clauses, [IF](#) structures, [WHILE](#) loops, the [TALLY](#) command, and in the [RULES](#) command where comparisons are equal or not equal. R:BASE stores the setting with the database. If CASE is set off, both uppercase and lowercase text are displayed for a comparison regardless of how you enter the text. For example, if you enter "case", you could find "Case" and "CASE".

1.20.8.17 CHECKPROP

Operating Condition

Syntax: SET CHECKPROP ON/OFF

Default: OFF

CHECKPROP displays or suppresses error message for PROPERTY and GETPROPERTY commands during processing. The PROPERTY/GETPROPERTY command errors can be displayed if a Component ID is not found or if a property value is invalid. The setting applies for command use in forms, reports, and labels. The errors are only displayed if CHECKPROP is set ON.

1.20.8.18 CLEAR

Operating Condition

Syntax: SET CLEAR ON/OFF

Default: ON

Mode: Single-user

SET CLEAR determines when R:BASE clears the internal buffers and transfers the data to disk.

When CLEAR is set on, the internal buffers are cleared and data is transferred to disk after each modification. Setting CLEAR on does not always make R:BASE automatically write each new or changed row to disk. For example, when you use a form, R:BASE writes the edits to disk when you finish using the form.

Set CLEAR off to write modified data to disk only when the buffer is full, a database is closed, or you exit R:BASE. If CLEAR is set off, repetitive modifications to a database can run faster, but you could lose all of your changes stored in the buffer if an accident, such as a fluctuation in power supply, occurs.

In multi-user mode, the CLEAR setting has no effect and always acts as though it is set on.

1.20.8.19 CLIPBOARD

Environment

Syntax: SET ClipBoard <TextString or Variable>

About the SET CLIPBOARD Command

Places the specified string onto the Windows clipboard, where it can be accessed from other programs.

EXAMPLE 01:

```
SET CLIPBOARD 'Here is text for the clipboard'
```

EXAMPLE 02:

Assuming you have already created a MS Word Document (TestDoc.DOC) or WordPerfect Document (TestDoc.WPD).

```
-- ClipBrd.RMD
CONNECT ConComp
SET VAR vAddressBlock TEXT = NULL
SET VAR vCustID INTEGER = 100
SELECT (Company+(CHAR(10))+CustAddress+(CHAR(10)) +
+CustCity+', '&CustState&CustZip+(CHAR(10))) +
      INTO vAddressBlock INDIC IvAddressBlock +
      FROM Customer WHERE CustID = .vCustID
SET CLIPBOARD .vAddressBlock
LAUNCH TestDoc.DOC
or
LAUNCH TestDoc.WPD
```

Once the MS Word or WordPerfect is launched and the document is opened, you could either use Edit > Paste or Ctrl+V to paste the windows clipboard text!

The resulting pasted block of text would look like:

```
Pc Distribution Inc.
3200 Westminster Way
Boston, MA 02178
```

Above example could be used in your application command file or in an EEP.

1.20.8.20 CMDHIST

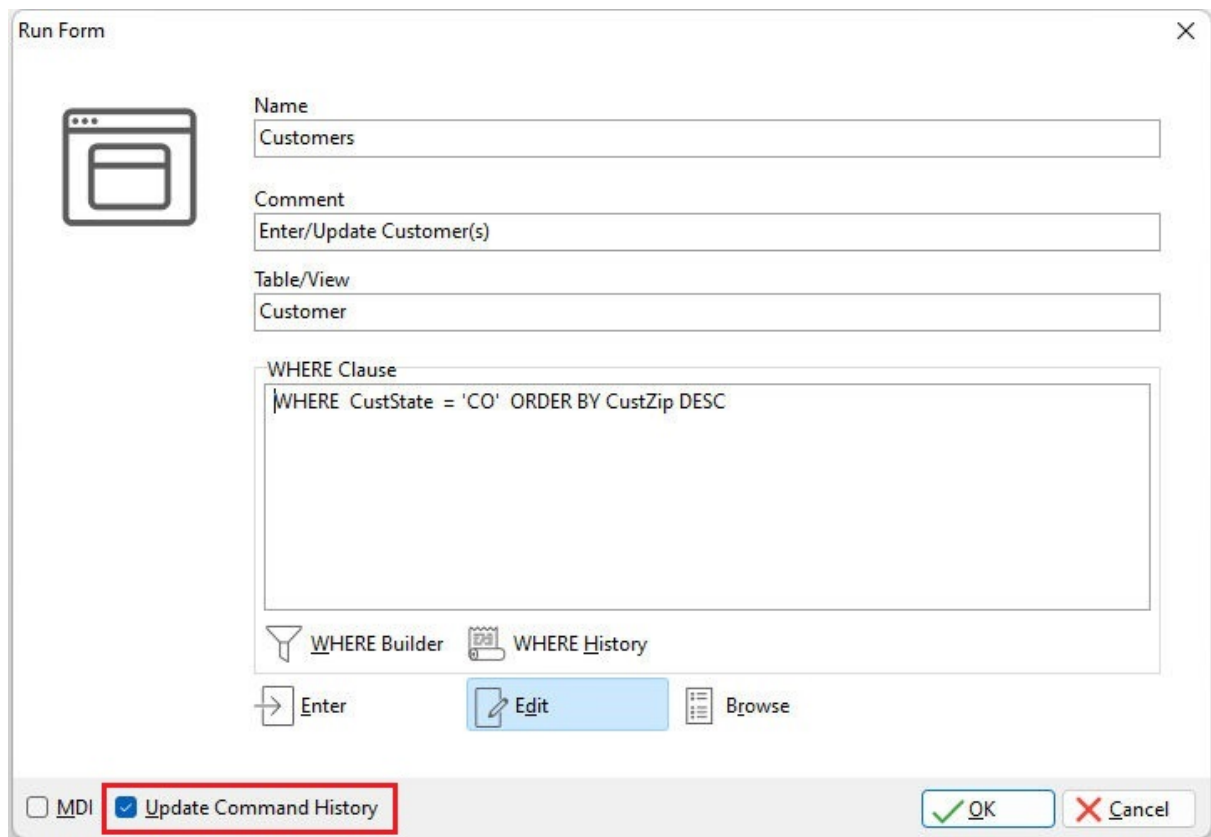
Operating Condition

Syntax: SET CMDHIST ON/OFF

Default: ON

Use the CMDHIST setting to control the status of the "Update Command History" check box when running a form or printing a report/label from the Database Explorer or the main toolbar. When the dialog is opened, a check is placed in the "Update Command History" check box, if CMDHIST is ON.

The "Update Command History" check box places the syntax used to execute the form/report/label within the R> Prompt Command History tool bar, to easily reference the command syntax at a later time.



The image shows a 'Run Form' dialog box with the following fields and controls:

- Name:** Customers
- Comment:** Enter/Update Customer(s)
- Table/View:** Customer
- WHERE Clause:** WHERE CustState = 'CO' ORDER BY CustZip DESC
- Buttons:** WHERE Builder, WHERE History, Enter, Edit, Browse
- Footer:** MDI (unchecked), Update Command History (checked), OK, Cancel

1.20.8.21 CMPAUSE

Operating Condition

Syntax: SET CMPAUSE ON/OFF

Default: OFF

CMPAUSE (Cascade Modal Pause) determines if R:BASE will use a local [PAUSE](#) dialog window for modal PAUSE displays instead of the global PAUSE form. This means multiple modal PAUSE dialog windows will appear on top of each other instead of prematurely closing the current modal PAUSE form.

An example would be when several PAUSE dialog will appear within a loop. CMPAUSE ensures the dialog messages will appear on top of one another.

1.20.8.22 COLCHECK

Operating Condition

Syntax: SET COLCHECK ON/OFF

Default: ON

Specifies if columns will be checked for consistency when making views.

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

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

1.20.8.23 COLOR

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET COLOR
SET COLOR FOREGRND color
SET COLOR BACKGRND color
SET COLOR BACKGRND (redvalue, greenvalue, bluevalue)

You can specify foreground and background colors for the DOS R> Prompt window. You can alter the R:BASE for Windows R> Prompt background and font color by adjusting the settings available from the main Menu Bar under "Settings" > "R> Prompt".

You can also specify colors using the SET command. For example, to change the background color to cyan, enter:

```
SET COLOR BACKGRND cyan
```

You can even select a custom color for the background using a combination of red, blue, and green values. For example, to change the background to orange, enter:

```
SET COLOR BACKGRND (255, 128, 64)
```

1.20.8.24 COMPATIB

Operating Condition

Syntax: SET COMPATIB ON/OFF

Default: ON

Compatibility with R:BASE Transactions

SET COMPATIB toggles COMPATIBILITY with R:BASE transactions on and off.

The first "compatibility" setting in R:BASE goes back to the 3.1 versions where it was used to allow concurrent access to a database from both 3.1 and 2.11 at the same time. Remember that database files back then were still "rbf" files. Once we went to the "rb1,rb2,..." files the setting was obsolete.

In 1997 when [Oterro](#) was first released, it supported a different scheme for managing transactions (when [TRANSACTIONS](#) are set on) than R:BASE itself used at that time. This new scheme used a different file to track transactions and had a larger allocated buffer size on the file to manage ongoing transactions. To make R:BASE compatible with Oterro when transactions were on it needed to support the new method, but it also needed to be able to support the older style that previous versions of R:BASE used (version 6.0 and older).

The first R:BASE version that could support the two methods was 6.1. When the "compatibility" setting is on, R:BASE will use the older "non-Oterro" style of transaction. When "compatibility" is off, the Oterro method will be used. If you never run with TRANSACTIONS ON then the compatibility setting does not have any effect.

1.20.8.25 CURRENCY

Operating Condition

Syntax: SET CURRENCY \$ PREF 2 B

SET CURRENCY sets the symbol, location, subunits, and format for currency values. R:BASE stores the setting with the database.

Changing the CURRENCY parameters affects all columns in the database that have a CURRENCY data type. You must enter the parameters in the following order: SYMBOL, PREF or SUFF, digits, and format; that is, even if you want to change the digits only, you must also enter the symbol and its position.

- **Symbol** (default \$)--A symbol is any ASCII character or string of one to four characters. You can include a space in place of one character at the beginning or end of the string, but if you do, enclose the string in quotation marks.
- **PREF and SUFF** (default PREF)--Specify the position of the symbol as before (PREF) or after (SUFF) the currency value. In the SET CURRENCY command, enter a space between the symbol and its position, PREF or SUFF.
- **Subunit digits** (default 2)--Indicates the number of digits from 0 to 16 to be displayed in a currency subunit. In the case of dollars, the subunit is cents, so the digits setting for dollars is 2. For example, setting digits to 3 will display currency values similar to these:

```
20.000,000DM
20,000.000
2,000.000
```

If you change digits when the database contains data, the new digit setting affects how R:BASE displays and uses the data already entered. For example, if you change the setting from 2 to 4, an existing value such as 1,234.00 becomes 12.3400.

- **Format** (default B)--Format specifies how R:BASE displays the thousands and decimal delimiters. *A*, *B*, *C*, *D*, and *E* specify how the thousands and decimal delimiter displays for values with CURRENCY, REAL, and DOUBLE data types. Before you change the format, you must change the DELIMIT character.

Delimiter Conventions for CURRENCY, REAL, and DOUBLE Values

Convention	Thousands Delimiter	Decimal Delimiter	Example
A	.	,	123.456.793,01
B	,	.	123,456,793.01
C	(blank)	,	123 456 793,01
D	N/A	.	123456793.01
E	'	.	123'456'793.01

For example, to display currency in two digits with a prefix of € (Euro) with the display format *A*, at the R> Prompt, enter:

```
SET DELIMIT=!
SET CURRENCY € PREF 2 A
```

This configuration displays 1,500 Euro and 25 cents in the format like this:

```
€ 1.500,25
```

If you set CURRENCY to the delimiter formats *A* or *C* without changing DELIMIT to a character other than the comma, the default setting for DELIMIT, R:BASE displays the following message: "Decimal character cannot be the same as DELIMIT."

In this case, use [SET DELIMIT](#) to change the delimiter to a less commonly used character such as an exclamation point; then set CURRENCY to the delimiter formats *A* or *C*.

1.20.8.26 DATE

Operating Condition

Syntax: SET DATE CENTURY value
 SET DATE YEAR value
 SET DATE MM/DD/YY (date sequence and format)
 SET DATE SEQ MMDDYY (date sequence)

SET DATE FOR MM/DD/YY (date format)

SET DATE sets the date sequence for entry, and format for display. A valid date can have up to 30 characters. R:BASE stores the setting with the database.

Use the SET DATE CENTURY *value* command to set the default century (the first two digits of a four-digit year). For example, if you enter a two-digit year and you want it to default to the twenty-first century, enter the following command:

```
SET DATE CENTURY 20
```

A year such as "25" would be stored as "2025."

Use the SET DATE YEAR *value* command to have two default centuries for dates entered, depending on the year. All years from 00 to (YEAR *value*-1) are stored with the next century (CENTURY *value*+1), and all years from *value* to 99 are stored with the default century (CENTURY *value*). For example, you can have all dates from the year 50 to 99 default to the twentieth century, and all dates from 00 to 49 default to the twenty-first century by entering the following commands:

```
SET DATE CENTURY 19
SET DATE YEAR 50
```

The SET DATE CENTURY 19 command sets the default century to 19. The SET DATE YEAR 50 command stores all years from 50 to 99 with the default century, 19. All dates from 0 to 49 are stored with the next century, 20. Therefore, years entered from 50 to 99 are stored as 1950 to 1999, and years entered from 00 to 49 are stored as 2000 to 2049.

Note: The DATE CENTURY and DATE YEAR options are only effective when the DATE SEQUENCE includes a two-digit year (MMDDYY, DDMMYY, etc.).

R:BASE accepts a date between January 1, 3999 BC and December 31, 9999 AD. You can set the date sequence and format separately. R:BASE displays the date based on the format. When setting the format to display numerals for the month, day, and year, use a separator such as the slash (/), hyphen (-), comma (,), or space (blank). For example, if you set the date format to MM/DD/YY and enter 061193, R:BASE displays 06/11/93.

You can also include text for the weekday and month in the date format to a maximum of 30 characters. Include WWW for a three-letter day abbreviation, WWW+ for the full day name, MMM for a three-letter month abbreviation, and MMM+ for the full month name. If the date format contains spaces or commas, enclose the format in quotes. For example, if the special character for QUOTES is set to the R:BASE default ('), the format 'MMM DD, YYYY CC' displays Jun 11, 1993 AD.

If you use YY in the date format, R:BASE displays only the last two digits of the year. To view dates in other centuries, use a date format with a four-digit year such as 'MM DD, YYYY'. If you use BC dates, add CC to the format. Dates entered with BC are shown with BC; otherwise, the date is shown with AD. For example, you could use the sequence MMDDYY and the format 'MMM DD, YYYY CC' to accept and display BC dates. If you enter '06 11 93BC,' R:BASE displays Jun 11, 0093 BC.

Example: Valid date formats, using June 11, 1993	
Date Format	Display
'MMM+ DD' MM/YY	June 11 06/93
'WWW the DD' 'WWW+', MMM+ DD, YYYY CC'	Sun the 11 Sunday, June 11, 1993 AD

You can omit the SEQ and FOR keywords to set both sequence and format in a single SET DATE command. For example, enter SET DATE MM/DD/YYYY to set both date sequence and format to a four-digit year.

Enter the date in any form as long as the sequence of M's, D's, Y's, and C's are in the same order defined for the date sequence. The display, however, is always exactly as defined by the DATE format.

If, for example, you set the date sequence to a four-digit year with SET DATE SEQ MMDDYYYY, set the date format to a two-digit year with SET DATE FOR MM/DD/YY, and later enter a two-digit year, R:BASE

will store and might display a date you do not expect. As the following table shows, if you enter 06/11/93, R:BASE stores the date as 06/11/0093 and displays 06/11/93.

How R:BASE stores and displays dates

DATE Sequence	Date Entered	Date Stored
MMDDYYYY	6/11/93	06/11/0093
MMDDYY	6/11/0093	06/11/1993
MMYY	6/11	06/01/1993
MMDD	6/11	06/11/1993
DDYY	6/11	01/06/1911
DDYY	11/93	01/11/1993

DATE Format (2-digit Year)	DATE Format (4-digit Year)
06/11/93	06/11/0093
06/11/93	06/11/1993
06/01/93	06/01/1993
06/11/93	06/11/1993
01/06/11	01/06/1911
01/11/93	01/11/1993

If the sequence is set to a four-digit year and the format is set to a two-digit year, R:BASE stores the date you enter, such as 06/11/95, as a four-digit year. As a result, if you use a WHERE clause to display rows that have dates greater than 06/11/95, R:BASE returns all rows greater than 06/11/0095.

The DATE format can affect date functions. For best results, set the format to the default MM/DD/YY and then use a date function.

1.20.8.27 DEBUG

Operating Condition

Syntax: SET DEBUG ON/OFF

Default: OFF

You can use SET DEBUG as follows:

1. Precede any R:BASE command you want to control with the DEBUG modifier in a command file.
2. Set DEBUG on to have R:BASE run the command; set DEBUG off to have R:BASE ignore the command.

Notes:

- When DEBUG is ON, R:BASE will display a warning if a SET VAR <NAME> <TYPE> declaration is executed when the variable already exists and the variable type changes. The feature is helpful to avoid accidentally clearing a variable when it is used elsewhere, especially in heavily nested code.
- With DEBUG ON, the #TABLEORDER variable value is generated to determine the column order used for a query.

1.20.8.28 DELIMIT

Special Character

Separates a character, string, or items in a list used in commands. Also used to separate repeatable parts of a command.

Syntax: SET DELIMIT=NULL
 SET DELIMIT=char
 (Use NULL to disable the special character.)

Default: ,

1.20.8.29 ECHO

Operating Condition

Syntax: SET ECHO ON/OFF

Default: OFF

SET ECHO displays or suppresses commands as they are processed from the current ASCII input device. Specify ON to display commands; specify OFF to turn off the command display.

Use SET ECHO as a debugging technique as you develop a command file. With ECHO set on, you can see the commands as they are processed when you run a command file. SET ECHO works only when the command file is an ASCII file; it will not display commands that were run from a binary procedure file.

Enter a SET ECHO ON command at the beginning of the program. Then, when the program runs, R:BASE displays each command as it is interpreted and, if needed, runs it. Sometimes the commands scroll on the screen faster than you can read, especially if they are read by R:BASE but not run. You can temporarily stop the display by pressing [Ctrl]+[Break], which stops the file from running, and restart it by pressing [Enter], or stop completely by pressing [Esc]. Set ECHO OFF to suppress command display.

Even more useful, you can direct output to a printer or a file before you set ECHO to on. Then, when the command file runs, the commands and any errors are saved either in printed form or in a file you can look at.

Echoing output can optionally be suppressed if a command starts with the "@" character, to improve the readability of an echoed output stream, for example, the echoing of the SET ECHO OFF command. Do not put any spaces in front of this character to make the output appear correctly.

When R:BASE runs the commands below in an ASCII file, you see the WRITE command display along with the message.

```
CLS
SET ECHO ON
WRITE 'This is a message'
SET ECHO OFF
```

To display the WRITE command along with the message, and without the SET ECHO OFF echo output, use the following:

```
CLS
SET ECHO ON
WRITE 'This is a message'
@SET ECHO OFF
```

1.20.8.30 EDITOR

Environment Setting

Syntax: SET EDITOR RBEDIT/filespec

Default: RBEDIT

The SET EDITOR setting allows you to specify the internal R:BASE Editor or some other text editor as your default text editor for R:BASE command files.

For example, if you wish to alter the default text editor to the external R:BASE Editor program, you would use the following syntax:

```
SET EDITOR C:\RBTI\RBEdit\RBEdit.exe
```

1.20.8.31 EOFCHAR

Operating Condition

Syntax: SET EOFCHAR ON/OFF

Default: ON

If EOFCHAR is set OFF, a control-Z character will not be appended to the end of output files.

1.20.8.32 EQNULL

Operating Condition

Syntax: SET EQNULL ON/OFF

Default: OFF

This Command determines whether or not NULL = NULL.

Compare these code samples:

```
SET VAR v1 TEXT = NULL
SET VAR v2 TEXT = NULL
```

```
SET EQNULL OFF
IF v1 = .v2 THEN
    -- will not be a hit
ENDIF
IF v1 <> .v2 THEN
    -- will not be a hit
ENDIF
IF v1 <> 'This' THEN
    -- will not be a hit (it used to be before this fix)
ENDIF
```

```
SET EQNULL ON
IF v1 = .v2 THEN
    -- will be a hit
ENDIF
IF v1 <> .v2 THEN
    -- will not be a hit
ENDIF
IF v1 <> 'This' THEN
    -- will be a hit
ENDIF
```

Before this fix, the comparison "IF v1 <> 'This' THEN" would be a hit with EQNULL set ON or FALSE when it should only be a hit when EQNULL is ON. This means that now "IF (.v1) <> 'This' THEN" and "IF v1 <> 'This' THEN" will both process the same way. In the past they would be different because of this problem.

In your code if you want the comparison of a NULL variable and a non-NULL constant to be a hit then you should run with EQNULL set ON.

1.20.8.33 ERROR MESSAGE

Special Condition

Syntax: SET ERROR MESSAGE Error# ON/OFF

Default: ON

SET ERROR MESSAGE Error# ON/OFF displays or suppresses a particular error message when a system error occurs.

You can selectively turn OFF any -ERROR- message(s) in your command file (very handy for debugging) by doing the following:

```
SET ERROR MESSAGE Error# OFF
```

To turn it back ON:

```
SET ERROR MESSAGE Error# ON
```

For example, to not see the:

-WARNING- No rows exist or the specified clause.

You can do the following:

```
SET ERROR MESSAGE 2059 OFF
```

This new feature has a limit of 50 -ERROR- numbers to set OFF and each one requires a separate command.

NOTE: Each turned OFF message must be turned back ON before turning it OFF again. If not, you'll get the error message.

1.20.8.34 ERROR MESSAGES

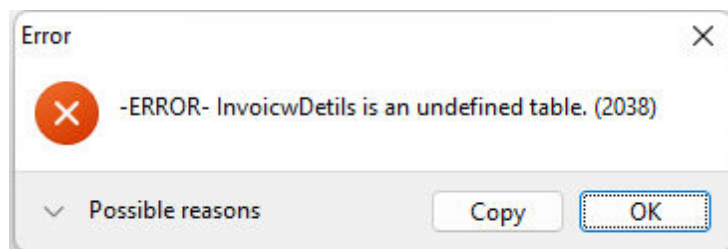
Operating Condition

Syntax: SET ERROR MESSAGES ON/OFF

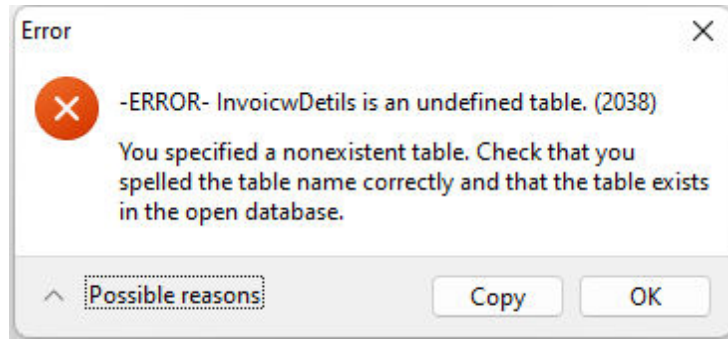
Default: ON

SET ERROR MESSAGES displays or suppresses an error message when a system error occurs. Specify off to suppress error and rule violation messages.

Error messages can also be suppressed after an initial error occurs. To do so, place a check within the "Suppress Error Messages" check box. After doing so, error messages will not be displayed for the instance of R:BASE until the program is restarted, or if the ERROR MESSAGES operating condition is set ON.



The Error Message dialog contains "Help" button to display possible reasons for the error, and a "Copy" button to capture the error.



1.20.8.35 ERROR VARIABLE

Operating Condition

Syntax: SET ERROR VARIABLE varname
SET ERROR VARIABLE OFF

SET ERROR VARIABLE defines an error variable to hold error message numbers. The variable name (*varname*) defines the variable R:BASE uses to hold R:BASE error codes. If set to off (the default), error variable processing is removed.

When an error occurs in a command file, R:BASE normally displays a system error message. SET ERROR enables a programmer to anticipate errors in command and procedure files and program the file to keep running even when an error occurs.

You must always set ERROR VARIABLE off, rather than clearing it with the [CLEAR VARIABLES](#) command.

R:BASE resets the error variable to zero as each command is successfully run. If an error occurs, the error variable is set to the error number value. To determine the error condition for any line, you must immediately check the value of the error variable or capture the error value in a global variable for later examination.

By checking the error variable for a non-zero value, you can detect (or trap) many errors and run a sequence of error-handling commands such as an error-recovery procedure. Once the error number is captured in an error variable, you can write error-handling command files to control a program's flow based on these errors (error values).

The error variable value is set for each command that is run, not each line in a command file. If you have placed multiple commands on a line, the last command's error value is placed in the error variable. A similar situation occurs for multi-line commands such as the subcommands you can use when loading a data block with the [LOAD](#) command. For example, a data block loaded with the LOAD command leaves the error variable with a value of zero because the END command runs successfully, whether or not the data is actually loaded.

Rule violations do not set the error variable to a non-zero value; they are not the same as errors recognized by R:BASE.

The command below defines *errvar* as the current error variable:

```
SET ERROR VARIABLE errvar
```

When a command is run, R:BASE sets the error variable *errvar* to the error code before anything else happens. The following command lines illustrate how to use *errvar* in a command file.

```
LABEL tryagain
DIALOG 'Enter the database name:' vdbname vendkey 1
CONNECT .vdbname
```

```

IF errvar <> 0 THEN
    WRITE 'Database not found.'
    GOTO tryagain
ENDIF

```

The first command establishes a label to return to, the second requests that the user enter the name of a database, and the third opens the specified database using the global variable defined by the [DIALOG](#) command.

The [IF...ENDIF](#) structure checks the error variable value. If the value is not zero (that is, if the database was not opened successfully), then it sends a message to the screen and passes control to the label *tryagain* so that the user is asked to enter the database name again.

You can also write a separate command file specifically designed to handle a variety of errors. In this case, the above code might look like this:

```

DIALOG 'Enter the database name:' vdbname vendkey 1
CONNECT .vdbname
SET VARIABLE verr1 = .errvar
IF verr1 <> 0 THEN
    RUN errhandl.cmd USING .verr1
ENDIF

```

This series of commands captures the error value in the global variable *verr1* so that it can be passed through the USING clause of the [RUN](#) command to an error-handling routine. The routine itself determines the nature of the error and how to take care of the problem.

You can use the [WHENEVER](#) command to run status-checking routines for SQL commands. **WHENEVER** uses the special R:BASE variable **SQLCODE**.

See also:

```

(CVAL('ERROR VARIABLE'))
(CVAL('LAST ERROR'))

```

1.20.8.36 ESCAPE

Operating Condition

Syntax: SET ESCAPE ON/OFF

Default: ON

SET ESCAPE allows you to use [Ctrl]+[Break] to escape or abort command file processing or database file access. Specify on to enable users to abort processing in the middle of command files, [WHILE](#) loops, and database access. Specify off to prevent users from prematurely aborting a command file or an application such as when R:BASE runs processes that create new tables (such as the relational commands) from within a command.

1.20.8.37 EXPLODE

Operating Condition (**R:BASE for DOS only**)

Syntax: SET EXPLODE ON/OFF

Default: OFF

Controls how DOS dialogs are displayed.

When EXPLODE is set on, dialog boxes are displayed in full size instantly. When EXPLODE is set off, dialog boxes are displayed in an expanding fashion from the center.

1.20.8.38 FASTFK

Operating Condition

Syntax: SET FASTFK ON/OFF

Default: OFF

This setting, when on, permits R:BASE to operate a foreign key index using a condensed index for maintaining that foreign key. If the foreign key is not used for retrieving data or linking columns, a complete index is unnecessary and actually inhibits speed. When set to on, R:BASE creates a condensed index for any existing foreign keys.

To switch to condensed indexes on existing foreign keys, you need to run a [PACK](#), [PACK KEYS](#), or [RELOAD](#) command with FASTFK on; these actions cause R:BASE to rebuild the database with condensed foreign key indexes.

Keep in mind, however, that you might need complete indexes on foreign keys where such indexes are needed for retrieving data. Retaining a separate index on columns used in foreign keys that link tables is preferred. Indexes are also needed on foreign keys that you use for selecting column values; therefore, use the [CREATE INDEX](#) command to explicitly create indexes for columns used in foreign keys in a database where FASTFK is set on.

The command [SHOW FASTFK](#) displays the FASTFK state and whether FASTFK is operational in the current database. For example:

```
SHOW FASTFK
(FASTFK ) ON Use fast Foreign Key (FK) structures on rebuild.
          OFF FASTFK setting for current database
```

Once you rebuild the keys in a database with the FASTFK setting on, SHOW FASTFK displays the following:

```
SHOW FASTFK
(FASTFK ) ON Use fast Foreign Key (FK) structures on rebuild
          ON FASTFK setting for current database
```

1.20.8.39 FASTLOCK

Operating Condition

Syntax: SET FASTLOCK ON/OFF

Mode: Multi-user and [STATICDB](#)

Set FASTLOCK on for faster multi-user performance while modifying data. With FASTLOCK on, R:BASE does not place a table lock on the table, allowing for greater throughput. A table lock is only needed to prevent structure changes.

FASTLOCK can only be set on when STATICDB is set on, and both FASTLOCK and STATICDB must be set on before the database is connected. Like other R:BASE database modes (SET MULTI and SET STATICDB), FASTLOCK requires all users to be connected with the same setting.

The following command lines set STATICDB and FASTLOCK correctly.

```
SET STATICDB ON
SET FASTLOCK ON
CONNECT concomp
```

See Also:

Environment Optimization

1.20.8.40 FEEDBACK

Operating Condition

Syntax: SET FEEDBACK ON/OFF

Default: OFF

This setting displays processing results when either calculating or editing rows.

With this setting ON, R:BASE displays the number of rows processed and the elapsed time to completion in a dialog window. Displays occur while in the Data Browser, when printing reports and labels, and when using certain R:BASE commands including:

1. [ALTER TABLE](#)
2. [AUTONUM](#)
3. [COMPUTE](#)
4. [CREATE INDEX](#)
5. [CROSSTAB](#)
6. [DELETE](#)
7. [INSERT](#)
8. The data transfer for the [JOIN](#) of two tables
9. [LOAD](#)
10. The data transfer for a [PROJECT](#) command
11. [SELECT](#)
12. SORTing a large record set
13. The data transfer for a [SUBTRACT](#) of two tables
14. [TALLY](#)
15. The [UNION](#) command
16. [UNLOAD](#)

When using any of the various commands at the R> Prompt with FEEDBACK set ON, the FEEDBACK system variables are generated to hold the processed row count and elapsed time for the command.

RBTI_RowsInserted	= 101010	INTEGER
RBTI_RowsDeleted	= 0	INTEGER
RBTI_RowsUpdated	= 0	INTEGER
RBTI_ElapsedTime	= 0:00:01.468	TEXT

By default the FEEDBACK message is always in the center of the screen. However, you may specify coordinates for the FEEDBACK message box, rather than being always in the center of the screen. Use the following PROPERTY commands to customize the location of the FEEDBACK message box:

```
PROPERTY APPLICATION FEEDBACK_POSITION_TOP 35
PROPERTY APPLICATION FEEDBACK_POSITION_LEFT 35

PROPERTY APPLICATION FEEDBACK_POSITION_RIGHT 200
PROPERTY APPLICATION FEEDBACK_POSITION_BOTTOM 200

PROPERTY APPLICATION FEEDBACK_POSITION_VERTICAL_CENTER ' '
PROPERTY APPLICATION FEEDBACK_POSITION_HORIZONTAL_CENTER ' '
```

To display the FEEDBACK results from the last INSERT, UPDATE, or DELETE, without displaying the FEEDBACK dialog, initiate the following PROPERTY command parameter:

```
PROPERTY APPLICATION GET_LAST_FEEDBACK ' '
```

1.20.8.41 FILES

Operating Condition

Syntax: SET FILES value

Range: 1 to 255 files

Default: 25

The FILES setting establishes the maximum number of files that can be opened where you return to the next statement of the calling file.

If the values of FILES was 10, then you would only be able to go "10" files deep with one file calling another where you plan to return to the calling file. If you call an 11th file, then you would no longer be able to return to number 1. Any [RETURN](#) from the 2nd file would go back to the keyboard since file 1 was dropped off the return list.

The maximum, depending on available memory, is 255.

1.20.8.42 FIXED

Operating Condition

Syntax: SET FIXED ON/OFF

Default: ON

Controls column width in [SELECT](#)

This controls whether R:BASE will automatically shrink column widths in SELECT commands.

1.20.8.43 FONT

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET FONT keyword

Default: OEM

The FONT setting changes the font used in the "R> Prompt" window.

You can choose from three settings, which are mono-spaced stock fonts used in Windows:

- System--fonts compatible with the system font in Windows.
- OEM--an IBM PC character set for IBM computers.
- Ansi--a fixed-pitch font based on the Windows character set. A Courier font is typically used.

1.20.8.44 HEADINGS

Operating Condition

Syntax: SET HEADINGS ON/OFF

Default: ON

SET HEADINGS displays columns with or without headings when you enter the [SELECT](#) and [TALLY](#) commands.

1.20.8.45 IDQUOTES

Operating Condition

Syntax: SET IDQUOTES

Default: Reverse Quote (`) Prior Versions: NULL

Controls the character that is used to set off object names.

IDQUOTES sets the character for enclosing object names, such as column or table names, in R:BASE and ODBC commands. The IDQUOTES setting is especially critical when using ODBC to connect to a non-R:BASE database which allows characters such as spaces in column or table names. The IDQUOTES setting is also important in some internal R:BASE processing.

Note: Older databases may default to NULL which is not ODBC or SQL compliant.

In general, setting IDQUOTES will have no effect on legacy applications as they will not, in most cases, know that this exists. The one exception to this is unloading data from a database that has IDQUOTES set, such as an current upgraded R:BASE database, and importing into a legacy database. If you are attempting to do this you should set your IDQUOTES to NULL.

The reverse quote is located under the tilde (~) character and to the left of the numeral 1, on most standard US Keyboards.

1.20.8.46 INDEXONLY

Operating Condition

Syntax: SET INDEXONLY ON/OFF

Default: ON

Sets a flag to disable "index only" data retrievals.

When INDEXONLY is ON, R:BASE is enabled to retrieve data using only data from the index or key structures and not the actual table rows. This is the default behavior.

If INDEXONLY is set to OFF, R:BASE must retrieve data from the actual rows of the tables. R:BASE will still use an index to speed finding rows if possible, but actual data must come from the rows and not the index. This is sometimes required when multi-column indexes are used, and not all columns in the index are used in the WHERE clause.

1.20.8.47 INSERT

Operating Condition

Syntax: SET INSERT ON/OFF

Default: ON

SET INSERT turns insert/overwrite on or off. Set INSERT on to use either the insert or overwrite mode. Pressing the [Insert] key when you have specified INSERT to be on toggles you between insert mode and overwrite mode. In insert mode, the cursor indicator is larger. Press the space bar to insert a space. Set INSERT off to use only the overwrite mode; the cursor indicator is smaller.

1.20.8.48 INTERVAL

Operating Condition

Syntax: SET INTERVAL value

Default: 5

Range: 0 to 9 tenths of a second

Mode: Multi-user

The SET INTERVAL command specifies the time to elapse before R:BASE retries the command that caused a conflict within the waiting period. Also, see [WAIT](#).

1.20.8.49 KEYMAP

Operating Condition

Syntax: SET KEYMAP keyname OFF
SET KEYMAP ALL OFF
SET KEYMAP keyname=remapped keys

You can define key maps with a single statement, which enables you to define key maps in command files. For example:

```
SET KEYMAP [SHIFT][F3] TO [F2]  
SET KEYMAP [ALT]M= [F2]
```

To return to the original key mapping, enter the following:

```
SET KEYMAP [ALT]M OFF
```

1.20.8.50 LAYOUT

Operating Condition

Syntax: SET LAYOUT ON/OFF

Default: ON

SET LAYOUT switches saving layouts on or off. When LAYOUT is set on, R:BASE saves the layout of data displayed in the Data Browser when you exit. R:BASE saves layouts for single tables only. The next time you display the table with the Data Browser, the layout of data will be as you previously arranged it.

When LAYOUT is set off, R:BASE does not save the layout of data and ignores any saved layouts. If you want to see a table displayed in its default format without changing the saved layout for it, set LAYOUT off before displaying the table.

1.20.8.51 LINEEND

Special Character

Syntax: SET LINEEND = value

DOS Default:  [Alt]+[0254]

Windows Default: ^

You can set an end of line ASCII character for NOTE and TEXT fields in forms, reports and the Data Browser. When you insert the character in those fields and then zoom in by pressing [SHIFT]+[F2] or print a report, you see lines break as established by the line end character. The default character for R:BASE database that migrated over the years from DOS versions is , the ASCII value 0254. The default character for R:BASE for Windows is the carat (^).

Note: If the line end character has been set to the currency character, R:BASE changes it to ASCII value 0254 when you connect to a database.

1.20.8.52 LINES

Operating Condition

Syntax: SET LINES value

Range: 0 to 32,767 lines

Default: 20

SET LINES sets the number of lines per page or screen when you use the [CROSSTAB](#), [DISPLAY](#), [DIR](#), [LIST](#), [OUTPUT](#), [SELECT](#), [LIST RULES](#), [SHOW VARIABLES](#), [TALLY](#), or [TYPE](#) commands. LINES does not

affect report generation; you can define the number of lines on a page for each report. Setting LINES to zero displays lines as continuous output.

1.20.8.53 LOCK

Operating Condition

Syntax: SET LOCK tbllist ON/OFF

Default: OFF

Mode: [Multi-user](#)

SET LOCK manually sets locks on or removes locks from tables specified in the list of tables (*tbllist*). Use the command whenever you want a procedure or transaction to have exclusive use of tables. Setting LOCK to off disables locks for each of the tables in *tbllist*.

In command or procedure files it is sometimes necessary to prevent access to a table or group of tables while certain operations are performed. Although R:BASE handles most locks automatically according to the command that is running, at times you might want more control over table locking. SET LOCK provides you explicit control over access to tables during processing by the commands that retrieve and update data.

If R:BASE cannot lock all the tables listed after SET LOCK, it issues a message saying that not all tables are available to be locked. R:BASE does not lock any tables unless it can lock all tables listed, and it sets an error code when SET LOCK fails.

Group the tables used into one SET LOCK command to avoid tying up needed resources. Be sure to issue the SET LOCK OFF command to remove the locks after processing is complete. Locks set with this command are cumulative. You need to issue one SET LOCK OFF command for each SET LOCK ON command that you have entered for a given table. The user who set the table locks must issue SET LOCK OFF; otherwise no other user can access the locked tables until the first user exits the database. The [CLEAR](#) TABLE LOCKS command may be used to clear all table locks, which must be used with [MULTI](#) set to OFF and while connected to the database.

[LIST](#) displays locked tables with a letter next to the table name for the type of multi-user lock.

- (L) for a local lock
- (R) for a remote lock
- (C) for a cursor lock
- (r) for a row lock

LIST tells you whether the lock is a row, cursor, local, or remote lock. Row and cursor locks are set by R:BASE as part of its internal concurrency control. A local lock is set by a SET LOCK command issued at the workstation that issued the LIST TABLE command. And a remote lock is set by a command that obtains a table lock and is issued from a workstation other than the workstation that issued the [LIST](#) [TABLE](#) command.

The list of locked tables may be acquired using the (CVAL('TABLELOCKS')) function. The list of locked views may be acquired using the (CVAL('VIEWLOCKS')) function.

The lock status for a table may be acquired using the GetLock parameter for the GETVAL function. The lock type on a table/view may be acquired using the GetLockType parameter for the GETVAL function.

The first command line below sets an exclusive lock on the *customer* table. The second command line sets additional exclusive locks on the *transmaster* and *transdetail* tables. These exclusive locks prevent access to the three tables by any user other than the one who issued the SET LOCK ON commands. The SET LOCK OFF command removes the locks on all three tables.

```
SET LOCK customer ON
SET LOCK transmaster, transdetail ON
SET LOCK customer, transmaster, transdetail OFF
```

1.20.8.54 LOOKUP

Operating Condition

Syntax: SET LOOKUP value

Default: 5

SET LOOKUP tells R:BASE how many form look-up expressions to store in memory. Storing a look-up expression in memory enables R:BASE to display data more quickly in a form. The number of look-up expressions you can specify depends on the memory available in your computer. R:BASE needs approximately 500 bytes of RAM for each look-up expression. SET LOOKUP does not affect master look-up expressions or pop-up menus in a form.

For example, when your form contains 10 look-up expressions and LOOKUP is set to 5, only the first five look-up expressions are stored in memory. R:BASE must retrieve/reevaluate the remaining look-up expressions. So that the form works faster, you can set LOOKUP to a higher value to store more look-up expressions in memory.

1.20.8.55 MANOPT

Operating Condition

Syntax: SET MANOPT ON/OFF

Default: OFF

MANOPT set to OFF disables the automatic table-order optimization that R:BASE performs when running queries. This gives maximum control over the order in which columns and tables are assembled in response to a query.

With MANOPT set to ON, R:BASE uses the order of the tables in the [FROM](#) clause and the order of the columns in the column list of the [SELECT](#) clause to construct the query.

When MANOPT is set on, the **#TABLEORDER** system variable stores the table join order, and the applicable indexed columns. To display the table order, use the following:

```
PAUSE 2 USING .#TABLEORDER
```

See Environment Optimization

1.20.8.56 MANY

Special Character

Sets the character for the many wildcard for R:BASE commands and clauses.

Syntax: SET MANY=NULL
SET MANY=char
(Use NULL to disable the special character.)

Default: %

1.20.8.57 MAXTRANS

Operating Condition

Syntax: SET MAXTRANS value

Range: 1 to 1295

Default: 201

Mode: Transaction Processing

SET MAXTRANS specifies the maximum number of users who can have the same database open concurrently with transaction processing on.

Only the first user to connect to a closed database can enter the MAXTRANS setting for that database. Enter the command before connecting to the database. If anyone else already has the database open, R:BASE displays a message telling you that your SET MAXTRANS command will have no effect on the database.

MAXTRANS can affect system performance. The higher the MAXTRANS setting, the more overhead the system must carry to process transactions across the network. Also, the more users who are actually entering transactions, the slower the system operates.

1.20.8.58 MDI

Operating Condition

Syntax: SET MDI ON/OFF

Default: OFF

Use SET MDI to control the "MDI" option when running a form, printing a report/label to the screen, running an external form, or opening a table/view with the Database Explorer "Use Dialog to Open Table/View" setting, from the Database Explorer or toolbar. When the dialog window is opened, a check is placed in the "MDI" check box if MDI is ON or displays no check if MDI is OFF.

MDI controls whether the window uses a modeless, or non-modal, window so the window can be left open while work continues elsewhere.

When using commands, MDI is available as a part of the syntax for the following:

[BROWSE](#)
[BROWSE USING](#)
[EDIT](#)
[EDIT USING](#)
[ENTER](#)
[LBLPRINT](#)
[PRINT](#)
[REF](#)

See Also:

[CLOSEWINDOW](#)
[SETFOCUS](#)
IFWINDOW
(CVAL('MDI'))

1.20.8.59 MESSAGES

Operating Condition

Syntax: SET MESSAGES ON/OFF

Default: ON

SET MESSAGES either displays or suppresses system messages. Set MESSAGES to off when ERROR is set on to display only error messages.

1.20.8.60 MIRROR

Operating Condition

Syntax: SET MIRROR <path>

Syntax: SET MIRROR OFF/DELETE

Default: OFF

SET MIRROR <path> maintains a duplicate copy of the database. This duplicate copy is created and maintained in the directory designated in *path*. The duplicate database will have the same name, therefore *path* must designate a backup directory. With this setting, all modifications to the original database are duplicated in the mirrored database. Be sure all users are mapped to the save drive letters. SET MIRROR OFF turns off mirroring of the database; SET MIRROR DELETE turns off mirroring and then deletes the duplicate database.

A database copy must be made initially to the designated mirror directory. [MULTI](#) must be set to OFF in order to apply the MIRROR setting.

To review a mirrored database and the duplicated content, the live and duplicate copies of the database must have MULTI and MIRROR set to OFF.

The following establishes a mirror database for the RRBW20 database to the R:\RRBW20\MirrorDB folder:

```
DISCONNECT
SET MULTI OFF
CONNECT RRBW20
SET MIRROR R:\RRBW20\MirrorDB

SHOW MIRROR
(MIRROR    ) R:\RRBW20\MirrorDB --Database MIRROR files location
```

1.20.8.61 MOUSE

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET MOUSE -1 to 100

Default: 30

Controls how DOS dialogs are displayed.

When [EXPLODE](#) is set on, dialog boxes are displayed in full size instantly. When EXPLODE is set off, dialog boxes are displayed in an expanding fashion from the center.

Controls mouse sensitivity

Used with R:BASE for DOS only. SET MOUSE controls the period of time in hundredths of a second in which the mouse registers a double click. Setting the time too low makes it impossible to double-click the mouse. A setting of -1 disables the mouse. To set the mouse speed each time you use R:BASE, include the SET MOUSE command in a startup file.

1.20.8.62 MULTI

Operating Condition

Syntax: SET MULTI ON/OFF

Default: OFF

SET MULTI sets Multi-User capability on or off when you next connect a database. This setting must be used while you are disconnected from a database.

1.20.8.63 NAME

Operating Condition

Syntax: SET NAME network identification

Default: USER***** (USER and 14 numerals for date and time)

Mode: Multi-user

SET NAME specifies a network identification for your system when you start R:BASE. NAME is text and can contain spaces.

NAME must be saved to the configuration file.

1.20.8.64 NAMEWIDTH

Operating Condition

Syntax: SET NAMEWIDTH value

Range: 4 to 128 characters

Default: 18

SET NAMEWIDTH controls the name width of a table, column, form, report, label, etc. that R:BASE directs to the printer, screen, or file when using the [BACKUP](#), [COMPUTE](#), [CROSSTAB](#), [DISPLAY](#), [LIST](#), [SELECT](#), [TYPE](#), [UNLOAD](#), or [WRITE](#) commands. The defined width value specifically controls the number of characters for the first column in the displayed list, whether it is the first column for a SELECT command, or the table names in a LIST TABLES command. Use the [WIDTH](#) setting to control the number of characters for the entire row/line of data.

Do not set the width to a number greater than the number of characters your printer can fit on a line; a typical page and computer screen display 80 characters. WIDTH does not affect report generation; each report defines the width of a data line.

The LIST TABLES command produces the below results when NAMEWIDTH is 18 and WIDTH is 79.

R>LIST TABLES

Name	Columns	Rows	Comments
BonusRate	3	7	Rates for Bonuses
Component	2	12	Component Identification Number and Description
CompUsed	2	22	Components Used in a Model
Contact	12	35	Customer Contact Information
ContactCallNotes	5	3	Contact Call Notes
Customer	17	30	Customer Information
DBAccess	1	1	Database Access
Departments	8	31	Departments
Employee	19	13	Employee Information
FormTable	1	1	Dummy Table for Forms
HourlyTemps	3	3	Temperature Data for Gauge

With the NAMEWIDTH value doubled to 36, the "Name" column width is increased. Notice that the "Comments" column is now wrapped.

R>SET NAMEWIDTH 36

R>LIST TABLES

Name	Columns	Rows	Comments
BonusRate	3	7	Rates for Bonuses
Component	2	12	Component Identification

			Number and Description
CompUsed	2	22	Components Used in a Model
Contact	12	35	Customer Contact Information
Customer	17	30	Customer Information

Increasing the WIDTH setting will display the "Comments" column without the wrapped characters.

```
R>SET WIDTH 120
```

```
R>LIST TABLES
```

Name	Columns	Rows	Comments
-----	-----	-----	-----
BonusRate	3	7	Rates for Bonuses
Component	2	12	Component Identification
Number and Description			
CompUsed	2	22	Components Used in a
Model			
Contact	12	35	Customer Contact
Information			
Customer	17	30	Customer Information
DBAccess	1	1	Database Access
Departments	8	31	Departments
Employee	19	13	Employee Information
FormTable	1	1	Dummy Table for Forms
HourlyTemps	3	3	Temperature Data for
Gauge			

1.20.8.65 NOCALC

Operating Condition

Syntax: SET NOCALC ON/OFF

Default: OFF

NOCALC suppresses or processes computed column expressions with the [UNLOAD](#) and [LOAD](#) commands.

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

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

NOCALC allow users to preserve original computed values when using UNLOAD/LOAD to move data, or when rebuilding a database with UNLOAD ALL.

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

R:BASE stores the NOCALC setting with the database.

1.20.8.66 NOTE_PAD

Operating Condition

Syntax: SET NOTE_PAD

Range: 0 to 100 (percent)

Default: 10

This setting allocates an additional percentage of storage space in NOTE columns to accommodate value increases (additional text), so that rows don't need to move to different disk locations. This increases performance by reducing disk reading.

The default setting is 10% of the row size, and can be set from 0 to 100%. Setting NOTE_PAD to 0% disables padding.

1.20.8.67 NULL

Special Character

Syntax: SET NULL -0- (1 to 4 characters)

Default: -0-

SET NULL sets the display symbol for null. You can use up to four characters. If you set null to a blank space, R:BASE does not display rows composed entirely of nulls with the [SELECT](#) command. R:BASE stores the setting with the database.

If you enter the following command, R:BASE displays a slash in the absence of data:

```
SET NULL /
```

1.20.8.68 NUMROUND

Operating Condition

Syntax: SET NUMROUND ON/OFF

Default: OFF

Use NUMROUND to round any excess scale digits for numeric data types.

The default value is OFF, which means that numeric values must follow the definition for the maximum number of digits to the right of the decimal (the scale).

If NUMROUND is ON, then as long as the number would round to a valid numeric with that precision and scale, R:BASE will accept the value. This means that if you entered 0.237 with NUMROUND set to ON, it will be allowed and the stored value will be 0.24.

1.20.8.69 ONELINE

Operating Condition (**R:BASE for DOS only**)

Syntax: SET ONELINE ON/OFF

Default: OFF

Controls text and note field wrapping.

When set to ON NOTE and TEXT fields will never wrap to the next line in Reports and [SELECT](#) commands. Instead they will be truncated at the end of the column.

1.20.8.70 PAGELOCK

Operating Condition

Syntax: SET PAGELOCK ON/OFF

Mode: Multi-user

Default: ON

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

The settings for PAGELOCK are:

- **ON** - R:BASE uses page locking or row locking as appropriate. When PAGELOCK is ON and two or more users are updating rows within the same page of data, R:BASE only lets the first user update rows--the other users are locked out until the first user's update has been completed.
- **OFF** - R:BASE uses a fast row-locking method where only row locking is applied with no page locking. When PAGELOCK is OFF, you can lock rows of data instead of locking a page of data. You increase multi-user performance when PAGELOCK is OFF. And even more so when [STATICDB](#) and [FASTLOCK](#) are on.

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

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

For the value of PAGELOCK to have any impact at all, [ROWLOCKS](#) must first be ON.

Technically, the efficient and fastest method for updating data in multi-user environment is to SET STATICDB ON, SET ROWLOCKS ON, SET FASTLOCK ON, and SET PAGELOCK OFF. This particular combination will result in the fewest contentions between users.

Notes:

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

Example for Testing:

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

See Also:

Environment Optimization

1.20.8.71 PAGEMODE

Operating Condition

Syntax: SET PAGEMODE ON/OFF

Default: OFF

Mode: Single- or Multi-user

Use PAGEMODE to design reports through a custom R:BASE program. With PAGEMODE on, you create a page of a report in memory, then send the report to a printer or file. PAGEMODE is an alternative method to creating reports through the Report Designer; the two methods work in different ways.

The [WRITE](#), [SHOW VARIABLE](#), and [SELECT](#) commands are used to "display" data on a virtual page. To determine the row location of the cursor on the virtual page, use the ISTAT function with the keyword PAGEROW after a SHOW VARIABLE command. (ISTAT('pagerow') does not work with the WRITE command.) The [DECLARE CURSOR](#) command is usually used for retrieving data for printing. You control form feeds (new pages) in your program by using the [NEWPAGE](#) or [OUTPUT SCREEN](#) command.

At the beginning of your program, set both [SET LINES](#) and [SET WIDTH](#) to accommodate the report's size. Then set PAGEMODE to ON. You cannot write to a line longer than the current LINES setting or wider than the current WIDTH setting. You also cannot change the LINES and WIDTH setting without setting PAGEMODE to off first.

To send printer control codes to a printer, use the CHAR function to define a variable containing the printer control codes. Then, that variable is sent to the printer using the SHOW VARIABLE or WRITE command. For example,

```
SET VARIABLE vLandscape = (CHAR(27) + CHAR+
                           (38) + CHAR(108) + CHAR+
                           (49) + CHAR(79))

WRITE .vLandscape
```

The printer control codes can be found in the user's manual for your printer. PAGEMODE allows you to:

- Produce multi-column reports.
- Design different styles for the pages of the report, such as a report with a personalized letter as the first page.
- Create reports from tables with many-to-many relationships that have two or more detail sections.
- Format odd and even pages differently throughout a report.
- Create reports wider than 255 columns and longer than 84 lines.
- Print headers and footers only on the last page or first page.
- Customize reports so break headers and footers are located on the same page.
- Place a different break header on every page or the same break header on every page.

The following command line enables PAGEMODE:

```
SET PAGEMODE ON
```

For more information on using PAGEMODE, please see Using PAGEMODE.

1.20.8.72 PARSESEMI

Operating Condition

Syntax: SET PARSESEMI ON/OFF

Default: OFF

Use PARSESEMI to control the parsing of semi colon characters in the processing of ampersand variables. When OFF, R:BASE ignores semi colons in ampersand variables. If PARSESEMI is set to ON then you can have semi colons inside an ampersand variable to stack multiple commands.

1.20.8.73 PASSTHROUGH

Operating Condition

Syntax: SET PASSTHROUGH ON/OFF

Default: OFF

When PASSTHROUGH is set on, [SELECT](#) statements are sent directly to a foreign data source and are not processed by R:BASE. In addition to SELECT, the underlying retrieval for the [PROJECT](#) command is a query that can also be passed directly to the server and the results put in the "returned rows" on the fly. If the original table had 30 columns and you only need one, the PASSTHROUGH ON only asked the server for one column, whereas when PASSTHROUGH is OFF, the entire row is built by the ATTACHED table and then the column asked for is sent to the calling program. Another command that can take advantage of this includes [INSERT](#) when coming from a SELECT on a server table.

PASSTHROUGH can also allow special syntax, such as non-SQL syntax, or syntax that is not supported by R:BASE. With PASSTHROUGH set to ON, you can use functions that are only defined on the server system in a SELECT command, because those functions get "passed through" to the server for processing.

About the PASSTHROUGH Setting

If PASSTHROUGH is set to ON and foreign database is connected with [SCONNECT](#), the following will happen with a SELECT command:

1. If the table is a regular R:BASE table, the command works normally
2. If the table is a foreign data source table already attached with [SATTACH](#), the query is sent directly to the ODBC server and the returned values are shown.
3. If the table is not known to R:BASE at all, as either an R:BASE table or a foreign data source table, the query is sent directly to the ODBC server. If the server cannot process the query, an error is returned. If the server can process the query, the SELECT will work.

Example:

```
CONNECT RRBYW20
SET PASSTHROUGH ON
SCONNECT NorthWind
SELECT * FROM dbo.shippers
```

OrderID	Customer	EmployeeID	OrderDate	RequiredDate
10249	TOMSP	6	07/05/1996 00:00:00	08/16/1996 00:00:00
10251	VICTE	3	07/08/1996 00:00:00	08/05/1996 00:00:00
10258	ERNSH	1	07/17/1996 00:00:00	08/14/1996 00:00:00
10260	OTTIK	4	07/19/1996 00:00:00	08/16/1996 00:00:00
10265	BLONP	2	07/25/1996 00:00:00	08/22/1996 00:00:00
10267	FRANK	4	07/29/1996 00:00:00	08/26/1996 00:00:00
10269	WHITC	5	07/31/1996 00:00:00	08/14/1996 00:00:00
10270	WARTH	1	08/01/1996 00:00:00	08/29/1996 00:00:00
10274	VINET	6	08/06/1996 00:00:00	09/03/1996 00:00:00
10275	MAGAA	1	08/07/1996 00:00:00	09/04/1996 00:00:00
10280	BERGS	2	08/14/1996 00:00:00	09/11/1996 00:00:00
10281	ROMEY	4	08/14/1996 00:00:00	08/28/1996 00:00:00
10282	ROMEY	4	08/15/1996 00:00:00	09/12/1996 00:00:00
10284	LEHMS	4	08/19/1996 00:00:00	09/16/1996 00:00:00

The data from the dbo.shippers table in the Northwind database is available, but the table has not been attached.

A SELECT... INTO .. syntax can also be used to place values into variables.

```
SELECT CustomerID,EmployeeID INTO vCustID INDI iv1, vEmpID INDI iv2 FROM dbo.Orders
WHERE OrderID = 10263
```

```
R>SHOW VAR
```

Variable	= Value	Type
vCustID	= ERNSH	WIDETEXT
vEmpID	= 9	INTEGER

Notes:

- When PASSTHROUGH is ON, only the SELECT command can work with a server table that is not SATTACHed. If you want to do INSERT, UPDATE, or DELETE to a server table that is not SATTACHed, then you must use the [SSQL](#) command.
- With PASSTHROUGH set to ON, you can actually use functions that are only defined on the server system in a SELECT command, because those functions get "passed through" to the server for processing. For example, assume you have an SQL Server table called dbo.Testing with an integer column named cInt , the following can be executed:

```
SELECT cInt, power(1.1,cInt) AS PW FROM dbo.Testing
```

This will use the SQL Server POWER function to raise 1.1 to the "cint" power and the results will come back to R:BASE. If PASSTHROUGH is OFF, this command generates an error because the "power" function is not defined in the R:BASE program.

1.20.8.74 PLUS

Special Character

Sets the character for the command line continuation character.

Syntax: SET PLUS=NULL

Syntax: SET PLUS=char
(Use NULL to disable the special character.)

Default: +

1.20.8.75 POSFIXED

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET POSFIXED ON/OFF

Default: OFF

Controls how the AT parameter works.

When you use *AT row,col* to position dialog, pause and other windows, the actual position depends on the current font size of the R> Prompt window. If you always want the calculation to use the 8x12 size of the OEM font then set POSFIXED to ON. If you want the dimensions of the current font to be used then set POSFIXED to OFF.

1.20.8.76 PRINTER

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET PRINTER printername

Controls the DOS printer

SET PRINTER specifies the printer for your system. The configuration file reads the printer file and sets the printer values as variables for the printer.

```
SET PRINTER epson.prd
```

This command specifies an Epson printer as the printer file named in the configuration file. The extension .PRD is not required.

1.20.8.77 PROCEDURE

The SET PROCEDURE command locks a procedure so it can be replaced.

**SET PROCEDURE *procname* LOCK [ON
OFF]**

Area: Stored Procedures & Triggers

Options

ON

Enables a lock

OFF

Disables a lock set by SET PROCEDURE or GET LOCK.

About the SET PROCEDURE Command

The SET PROCEDURE works like the [GET LOCK](#) command without retrieving the stored procedure into an ASCII file. ON sets the lock; OFF releases the lock placed by the SET PROCEDURE or the GET command.

When a procedure is locked, only the user placing the lock can replace the procedure or remove the lock. The NAME setting is used for identification of the user.

1.20.8.78 PROGRESS

Operating Condition

Syntax: SET PROGRESS ON/OFF

Default: OFF

This setting displays processing results when building indexes, packing or reloading a database. With this setting on, R:BASE displays the process being performed, the overall progress, and the completion percentage of each.

1.20.8.79 QUALCOLS

Operating Condition

Syntax: SET QUALCOLS n

Default: 10

QUALCOLS specifies the number of qualkeys to assign to SQL attached tables.

When attaching external tables by selecting "Utilities" > "Attach SQL Database Tables" from the menu bar, or using the [SATTACH](#) command (without the "USING ALL/collist" keywords), the QUALCOLS setting is used to assign what columns uniquely identifies a row.

If a primary key or unique key was not found for the table being `SATTACH`ed, and the `USING` collist clause was not used to specify what columns uniquely identifies a row, then R:BASE assigns primary and unique key qualkeys for the attached table. R:BASE assigns a set of columns to identify the rows starting with the first column of the table. The number of columns used is limited by the value for `QUALCOLS`.

The following command line sets `QUALCOLS` to 5:

```
SET QUALCOLS 5
```

See Also:

Foreign Data Sources and ODBC

1.20.8.80 QUOTES

Special Character

Sets the character for quotation marks. This character is used around all text strings.

Syntax: `SET QUOTES=NULL`
`SET QUOTES=char`
(Use `NULL` to disable the special character.)

Default: `'`

1.20.8.81 RBADMIN

Operating Condition

Syntax: `SET RBADMIN ON/OFF`

Default: `OFF`

The `RBADMIN` setting is used for [RBAAdmin](#), the R:BASE Network Database Administrator utility.

In order for `RBAAdmin` to disconnect users from the database, this setting must be set to `ON`. All users, whether their setting for `RBADMIN` is `ON` or `OFF` will be seen within `RBAAdmin`. It is recommended that this setting be added to a database application startup file for ease of implementation with `RBAAdmin`. When `RBADMIN` is set `ON` and connections are made to the database, a hidden binary file will be created in the database directory. The name of the file is unique to each database; consisting of `"RBAAdmin_" + dbname + ".bin"`. The binary file for the `ConComp` sample database, with `RBADMIN` set `ON`, would be `"RBAAdmin_ConComp.bin"`.

1.20.8.82 RDEBUG

The R:BASE Engine `RDEBUG` setting is available to create a log file to help understand possible issues when running R:BASE commands and using ODBC with foreign data sources. The log file will contain Engine Functions as R:BASE establishes and frees connections, executes SQL statements, retrieves data and values, controls transactions, and handles data and values.

To enable debugging, add the `RDEBUG` setting to your R:BASE product configuration file, which creates a log file of the R:BASE Engine Functions.

The following provides the supported use of the `RDEBUG` setting in the configuration file:

01. - Debugging is off

```
RDEBUG OFF
```

02. - Debugging is on, where a log file is created in `C:\`

```
RDEBUG ON
```


03. - Debugging is on with a file path and name. For the below, the rbengine.log file is created in the C:\Temp\RDEBUG\ folder.

```
RDEBUG ON C:\Temp\RDEBUG\rbengine.log
```

After the DEBUG setting has been turned ON or OFF, R:BASE must be restarted in order for the setting to be recognized.

Important: The debug setting and logging adds overhead to the R:BASE engine and performance will decrease. After logging has been captured for a desired event where an issue occurs, the debug setting should be set to OFF in the configuration file.

When RDEBUG is OFF the log file may remain in the configuration file.

```
RDEBUG OFF C:\Temp\RDEBUG\rbengine.log
```

The default location for the R:BASE configuration files is in "C:\Users\Public\RBTI", with RBENGINE11.CFG used for R:BASE 11.

1.20.8.83 RECYCLE

Operating Condition

Syntax: SET RECYCLE ON/OFF

Default: OFF

If RECYCLE is ON, when adding new rows require a new block from file 2, a new routine is called which searches for a suitable unused block rather than always adding a new block to the end of the file

The criteria for such a block are:

- No other table uses it
- The block is further down the file than the current last block of the table

PROS

If a suitable block is found, the block will be allocated to the table requiring the additional space and File 2 will not grow as a result of this allocation. The main benefit of using RECYCLE is that the growth of File 2 will be minimized. Over time this can add up to significant savings on disk space and backup media.

CONS

Since a new routine is being called to search for a suitable block, there will be a slight performance penalty. The penalty will only be incurred when an [INSERT](#) requires a new block.

Considerations

For RECYCLE to be effective, all users should have the setting ON. SET RECYCLE is best used in the configuration file. RECYCLE will only have an impact when used in conjunction with [PACK TABLE](#). Dead space in File 2 must first be freed up before it can be reused. RECYCLE will not be of benefit if your database does not end up with lots of deleted rows over time, providing the opportunity to recover dead space.

Conclusions

Periodic use of PACK TABLE tablename in conjunction with RECYCLE ON will retard File 2 growth and reduce fragmentation. Use of [PACK INDEX](#) FOR tablename will keep the index statistics fresh and query optimization results maximized. The need for planned downtime will be reduced.

1.20.8.84 REFRESH

Operating Condition

Syntax: SET REFRESH value

Range: 10 to 65535 seconds

Default: 0

Mode: Multi-user

SET REFRESH specifies how often R:BASE refreshes the form or the Data Browser, and displays current data. It also automatically recalculates lower tables in forms. Specify zero to turn the setting off. When REFRESH is set off, R:BASE tells you of edits when you save or delete a row.

1.20.8.85 REVERSE

Operating Condition (**R:BASE for DOS ONLY**)

Syntax: SET REVERSE ON/OFF

Default: ON

SET REVERSE ON displays data-entry fields in reverse video in forms. R:BASE stores the setting with the database.

1.20.8.86 ROWLOCKS

Operating Condition

Syntax: SET ROWLOCKS ON/OFF

Mode: Multi-user

Default: ON

R:BASE uses row-level locking in a multi-user environment. This command causes R:BASE to lock only the required row for the current command instead of locking the entire table. For example, if multiple users are modifying the same table using the [UPDATE](#) command, R:BASE locks only the rows affected by each UPDATE. When ROWLOCKS is set off, R:BASE sets table locks during each UPDATE, regardless of how many rows are affected.

See Also:

Environment Optimization

1.20.8.87 RULES

Operating Condition

Syntax: SET RULES ON/OFF

Default: ON

SET RULES determines whether R:BASE checks data against all existing rules during data entry and modification when you use the [EDIT](#), [EDIT USING](#), [ENTER](#), [INSERT](#), [LOAD](#), or [UPDATE](#) commands, or the [import/export](#) utility.

Set RULES off to direct R:BASE to ignore all rules when rules are not defined for a table, you are archiving data, or you are transferring data into another database. This speeds up processing because R:BASE normally checks the SYS_RULES table even if no rules are defined for a table. R:BASE does, however, check each entry against the data type of the column regardless of the RULES setting. If the database is protected by a database owner's user identifier, R:BASE does not accept the SET RULES command until you enter the owner's user identifier.

1.20.8.88 SCRATCH

Operating Condition

Syntax: SET SCRATCH ON / OFF / TMP / <path>

Default: TMP

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

- SET SCRATCH ON to store temporary sort files on the database drive and directory.
- SET SCRATCH OFF to store temporary files on the current drive and directory.
- SET SCRATCH TMP to store temporary files in the Windows TEMP directory.
- SET SCRATCH <path> provides the path to the location where temporary files are stored.

You can use the SCRATCH command in the configuration file so that the setting is made prior to launching R:BASE.

About SCRATCH TMP

By default, R:BASE configuration files include the TMP for SCRATCH setting.

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

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

1.20.8.89 SELMARGIN

Operating Condition

Syntax: SET SELMARGIN value

Range: 0 to the width of your screen

Default: 0

Use SELMARGIN to set the left margin for displaying the results of a [SELECT](#) command. The default for SELMARGIN is 0 (zero), which sets the margin to column 2. Use SELMARGIN when an ASCII file requires a predefined position or when data has a required starting point.

Note: Setting SELMARGIN to 0 (zero) or 2 begins the left margin at column two.

1.20.8.90 SEMI

Operating Condition

Syntax: SET SEMI ON/OFF

Default: OFF

Use SEMI to set the semicolon (;) key as the command terminator instead of the [Enter] key.

When SEMI is set on, you can enter multiple command lines without a continuation symbol. Also, the semicolon runs commands created for other SQL databases, such as SQL Server.

Note: When SEMI is set on, all commands, including [EXIT](#), must be followed with a semicolon.

The following command line sets SEMI off:

```
SET SEMI OFF;
```

1.20.8.91 SEMI (Special Character)

Special Character

Sets the character for the command separator.

Syntax: SET SEMI=NULL
 SET SEMI=char
 (Use NULL to disable the special character.)

Default: ;

1.20.8.92 SERVER

Operating Condition

Syntax: SET SERVER ON/OFF

Default: ON

SET SERVER sets messages from a foreign data source on or off. When SERVER is set on, messages from the foreign data source are displayed.

1.20.8.93 SHORTNAME

Operating Condition

Syntax: SET SHORTNAME ON/OFF

Default: OFF

Alters the display format of the directory contents, where the file names are listed.

With SHORTNAME set to ON, the [DIR](#) command lists the file name, extension, size in bytes, and the date and time files were last modified, only listing the contents in the traditional format.

Example 1:

The following command could be used to review the database files placed in a temp folder. The display uses SHORTNAME set to OFF.

```
R>DIR *.RX?
```

```
Volume in drive C is Acer
Volume Serial Number is 4060-5572
```

```
Directory of C:\Temp\
```

```
17.11.11    04:56 PM           48,458 RBTIDATA.RX1
17.11.11    04:56 PM      3,112,960 RBTIDATA.RX2
17.11.11    04:56 PM      770,048 RBTIDATA.RX3
17.11.11    04:56 PM      12,288 RBTIDATA.RX4
25.08.11    01:38 PM       73,146 RRBW20.RX1
25.08.11    01:38 PM    1,310,720 RRBW20.RX2
25.08.11    01:38 PM     180,224 RRBW20.RX3
25.08.11    01:38 PM    3,964,928 RRBW20.RX4
            8 File(s)       9,472,772 bytes
            0 Dir(s)      41,285,623,808 bytes free
```

Example 2:

The following command could be used to review the database files placed in a temp folder. The display uses SHORTNAME set to ON.

```
R>DIR *.RX?
```

```
Volume in drive C is Acer
Directory of C:\Temp\
```

```
RBTIDATA RX1      48458 17.11.11   04:56p RBTIDATA.RX1
RBTIDATA RX2     3112960 17.11.11   04:56p RBTIDATA.RX2
RBTIDATA RX3      770048 17.11.11   04:56p RBTIDATA.RX3
RBTIDATA RX4      12288 17.11.11   04:56p RBTIDATA.RX4
RRBYW20 RX1       73146 25.08.11   01:38p RRBYW20.RX1
RRBYW20 RX2     1310720 25.08.11   01:38p RRBYW20.RX2
RRBYW20 RX3      180224 25.08.11   01:38p RRBYW20.RX3
RRBYW20 RX4     3964928 25.08.11   01:38p RRBYW20.RX4
           8 File(s)             9472772 bytes
           8 Dir(s)             41285361664 bytes free
```

1.20.8.94 SINGLE

Special Character

Sets the character for the single wildcard for R:BASE commands and clauses.

Syntax: SET SINGLE=NULL
 SET SINGLE=char
 (Use NULL to disable the special character.)

Default: _

1.20.8.95 SORT

Operating Condition

Syntax: SET SORT ON/OFF

Default: OFF

SET SORT sets the sort optimizer on or off. When set on, R:BASE sorts the minimal amount of data for large tables and recombines the sorted data with the unsorted rows using the minimum amount of disk space. Set SORT on when a sort fails.

SORT is only to be used when displaying a column, or columns, which are not indexed. Otherwise, no results will be displayed.

1.20.8.96 SORTMENU

Operating Condition

Syntax: SET SORTMENU ON/OFF

Default: ON

SORTMENU causes all data dictionary menus to be in alphabetical ascending order, including all pop-up menus that display tables, forms, views, labels, and reports. Menus with column names and values remain unsorted in their original order.

1.20.8.97 STATICDB

Operating Condition

Syntax: SET STATICDB ON/OFF

Mode: Multi-user

Default: OFF

SET STATICDB activates a read-only schema mode. A user who first connects to a database with STATICDB set to on engages that database to operate in a read-only schema mode whereby any user must have their STATICDB setting on in order to connect to that database.

The effect of having STATICDB set on is that no schema changes can occur during that connection session.

Schema Reading Mode with SET STATICDB

In a multi-user environment, R:BASE must be aware of any schema modifications made by a user. Being aware of schema modifications, which is necessary for database integrity, entails constantly re-reading schema information.

Because many data-processing operations do not make frequent schema changes to the database, such a re-reading unnecessarily slows down processing.

The command [SET STATICDB ON](#) instructs R:BASE to prevent any schema modifications, therefore, R:BASE does not read schema information. By default, STATICDB is set off. SET STATICDB OFF forces a re-read of schema information before running any command.

When connecting to a database with STATICDB set on, R:BASE displays the "Database Schema is Read-Only." message. When STATICDB is set on, any changes made to add new tables or views results in a temporary table/view. The table/view will be dropped when disconnecting from the database. The [LIST TABLE](#) command will display the temporary table and view names with a "(T)" next to the name.

Also, the following R:BASE commands are not allowed when STATICDB is set on.

ALTER TABLE*	DROP*
COMMENT ON*	RBDEFINE*
CREATE INDEX	RESTORE

* Access is not allowed to tables created prior to database connection.

No ALTER, DROP, CREATE, SCONNECT, SATTACH, SDETACH, ATTACH, and DETACH commands will have any effect. You can issue CREATE, ATTACH, and SATTACH commands but they will only create temporary tables or views. Similarly, PROJECT will create temporary tables with or without the TEMP keyword. Because the database schema is protected from changes, R:BASE doesn't need to worry that a user will change the schema in the middle of a series of commands.

The [UNLOAD ALL](#) command does not act upon temporary tables. You can, however, unload individual temporary tables with the UNLOAD command.

If a user has STATICDB set off, that user cannot connect to a database being used by a user who has STATICDB set on. All users accessing the same database must have a matching STATICDB setting.

You can find out what the current STATICDB setting is with the SHOW STATICDB command. You can also capture the current STATICDB setting as a value with the CVAL function:

```
SET VAR vcval = (CVAL('STATICDB'))
```

See Also:

Environment Optimization

1.20.8.98 TIME

Operating Condition

Syntax: **SET TIME SEQUENCE HHMMSS** (time sequence)
 SET TIME FORMAT HH:MM:SS (time format)

Default: **SET TIME SEQUENCE HHMMSS** (time sequence)

SET TIME FORMAT HH:MM:SS (time format)

SET TIME sets the time entry sequence and output format, using up to 20 characters. Set TIME entry sequence and display format in separate commands. Use H to specify hours, M for minutes, S for seconds, and .SSS for thousandths. TIME can be specified of up to thousandths of a second. R:BASE stores the setting with the database.

The keyword SEQ (sequence) sets the entry sequence such as HHMMSS while the keyword FOR (format) sets the display format. For example, the format HH:MM:SS can display 14:30:20. R:BASE displays midnight (24:00) as 0:00. You can also include AP to display time on a 12-hour clock. In the previous example, if you change the format to HH:MM:SS AP, R:BASE displays 2:30:20 PM. If the format contains spaces or commas, enclose the format in quotes. Enter the hours, minutes, and seconds in the order SEQ specifies.

Example: Valid Time Formats using two thirty and twenty seconds, p.m.

<u>Time Format</u>	<u>Displays</u>
HH:MM:SS	14:30:20
HH:MM:SS AP	2:30:20 PM
HH/MM/SS	14/30/20
HH-MM-SS AP	2-30-20 PM
HH:MM:SS.SSS	14:30:20.245
HH:MM:SS.SSS AP	2:30:20.245 PM

It is required that the order of items in the sequence must be in the same order as the items in the format. If H comes before M in the sequence, then H needs to come before M in the format as well.

It is supported to have the "SS.SSS" in the format, and omit the values from the sequence.

TIME can affect time functions. For best results, first set TIME to the default HH:MM:SS and then use the time functions.

1.20.8.99 TIMEOUT

Operating Condition

Syntax: SET TIMEOUT value

Range: 0 to 1440

Default: 0

Use TIMEOUT to shut down an inactive R:BASE session and exit to the operating system after a set amount of time passes. A countdown is always running if a TIMEOUT is set. The countdown will reset when keyboard or mouse activity is detected or if commands are running where table rows are being processed. This is a useful feature for automatically disconnecting idle R:BASE sessions for scheduled database maintenance.

The default for TIMEOUT is 0 (zero), which does not activate a countdown. TIMEOUT is set in minutes (not seconds), and all workstations must set TIMEOUT separately.

The following command line will exit a user to Windows after the user's workstation is inactive for one hour:

```
SET TIMEOUT 60
```

When a TIMEOUT occurs, a routine in the form of a command file, [RUN SELECT](#), or [CALL](#) to a stored procedure can be executed. However, the routine cannot include a [DIALOG](#), [FILLIN](#), or [PAUSE](#) command, or expect a keystroke. To execute a routine when a TIMEOUT occurs, the command file name or single-line command (RUN, CALL, etc.) may be passed to R:BASE with the **RBTI_TIMEOUT** variable.

Examples:

Command file example:

```
SET VARIABLE RBTI_TIMEOUT TEXT = 'C:\CustDB\CleanUp.rmd'
```

RUN SELECT example:

```
SET VARIABLE RBTI_TIMEOUT TEXT = 'RUN SELECT CmdData FROM IntrnlCmd WHERE CmdName =  
'TimeOutCode''
```

Stored procedure example:

```
SET VARIABLE RBTI_TIMEOUT TEXT = 'CALL TimeOutCode()'
```

Notes:

- The RBTI_TIMEOUT command file must end with a [RETURN](#) command.
- The TIMEOUT command will close ANY and ALL open forms, designers, and editors WITHOUT saving the changes which have been made since the last save. It is the responsibility of the developer and end-user to implement proper coding and/or behavior to eliminate unexpected shutdowns without saving the changes. The TIMEOUT command will disconnect from the currently opened database (if applicable) before terminating the R:BASE session.

1.20.8.10TJOURNAL

Operating Condition

Syntax: SET TJOURNAL ON/OFF

Default: OFF

Mode: Transaction Processing

SET TJOURNAL toggles journaling of commands that modify data when transaction processing is on.

The transaction journal log file is located in the same directory as the database. The file name consists of the database name followed by "_TJournal" with ".LOG" as the extension. For example, the RRBYW20 sample database would have a journal file with the name RRBYW20_TJournal.LOG.

1.20.8.10TOLERANCE

Operating Condition

Syntax: SET TOLERANCE value

Default: 0

SET TOLERANCE sets the tolerance for comparisons between numbers with REAL and DOUBLE data types. The default tolerance of 0 means that the match must be exact to six digits of accuracy for REAL numbers and to 15 digits of accuracy for DOUBLE numbers. R:BASE stores the setting with the database.

The following command specifies a tolerance of .1 when testing column values. If the tolerance is .1 and the [WHERE](#) clause specifies *colname* = 100, then values between 99.9 and 100.1 are valid. If you set the tolerance to one, the values between 99 and 101 are valid.

```
SET TOLERANCE .1
```

1.20.8.10TRACE

Operating Condition

Syntax: SET TRACE ON/OFF

Default: OFF

SET TRACE ON will execute TRACE (Interactive Command File Debugger) inside a command file to trace a block of code as defined.

Example 01:

TRACE filename.ext (typical command line option)

Example 02: (in a command file)

your code here

SET TRACE ON (this will start the trace within a command file)

Your code here ...

SET TRACE OFF (this will stop the trace within a command file)

Your remaining code here ...

Notes:

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

1.20.8.10:TRANSACT

Operating Condition

Syntax: SET TRANSACT ON/OFF
SET TRANS ON/OFF

Default: OFF

Mode: Transaction Processing

SET TRANSACT toggles transaction processing on and off. When transaction processing is set on and [AUTOCOMMIT](#) is set off, all commands entered after one [COMMIT](#) or [ROLLBACK](#) command until the next comprise a transaction. The commands in a transaction are executed as they are entered, but changes to the data and database structure are not made permanent until you enter COMMIT (or exit the database). You can undo all changes in the transaction by entering ROLLBACK.

When transaction processing is on and AUTOCOMMIT is also on, each command that is executed successfully is treated as a transaction and made permanent. ROLLBACK has no effect when AUTOCOMMIT is on.

Only the first user to connect to a closed database can enter the TRANSACT setting for that database. Enter the command before connecting to the database. If anyone else already has the database open, R:BASE displays a message telling you that your TRANSACT setting must match that of the open database before you can connect. Transaction processing is either on for all users or off for all users in a given database.

1.20.8.10:UINOTIF

Operating Condition

Syntax: SET UINOTIF ON/OFF

Default: ON

Controls the automatic user interface updates at the Database Explorer to refresh the list of tables, views, stored procedures, forms, reports, and labels.

Setting UINOTIF to OFF within an application will improve the performance. UINOTIF is not stored within the database files, and is specific to the current R:BASE session. UINOTIF is read from the configuration file when R:BASE is launched. If the setting is change during a session, it will only impact that session,

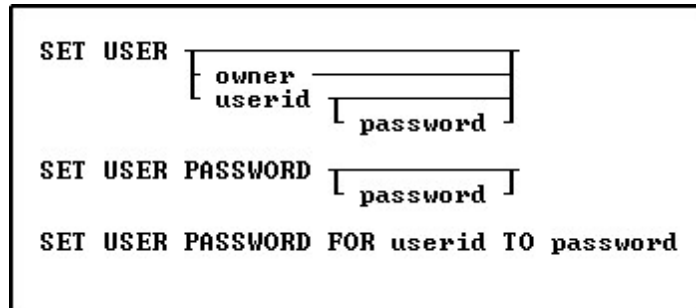
unless the configuration file is updated, in which case the new value will be reflected when a new R:BASE session reads the updated configuration file.

See Also:

Environment Optimization

1.20.8.10 USER

Use the SET USER command to create users, change the password for a user, or run R:BASE with a user identifier and password, if one has been set up.



Options

FOR userid

Specifies a user identifier. For a value with spaces, the userid must be enclosed in quotes.

owner

Specifies the database owner name.

password

Creates a new password. Enter NONE to remove an existing password.

PASSWORD

Specifies or changes the password for the current user identifier.

TO password

Creates a new password. Enter NONE to remove an existing password.

userid

Specifies a user identifier. For a value with spaces, the userid must be enclosed in quotes.

About the SET USER Command

Passwords are specific to user identifiers and databases and are not required by R:BASE; however, once a password is set up, R:BASE prompts for the user's password every time the user connects to the database or issues a user identifier. User identifiers have a maximum length of 36 characters. Passwords have a minimum length of three characters and maximum length of 36 characters.

To run R:BASE with your user identifier then connect to the database, enter the following command line:

```
SET USER <Userid>
```

You can also enter the following command line to run R:BASE with your user identifier:

```
SET USER
```

R:BASE displays a dialog box and prompts you for your user identifier.

When a password has been set up for a user identifier, R:BASE prompts for the password after the correct user identifier has been entered.

Note: When a user enters a user identifier or password in a dialog box, the user identifier is not displayed on screen.

To add or change a password, connect to the database with the user identifier and enter the following command line:

```
SET USER PASSWORD
```

R:BASE prompts for the user's identifier, then prompts for the password. A user can cancel a password by entering NONE.

If the database owner is the current user, the database owner can assign him/herself a password using the SET USER PASSWORD command; however, if the database owner forgets the assigned password, the password cannot be found or changed.

As the database owner, to change a user's password, connect to the database and enter the following command line:

```
SET USER PASSWORD FOR <Userid> TO <Password>
```

Enter your current password when R:BASE prompts you for it, then when R:BASE prompts you for a new password, enter NONE.

Note: A user's password is revoked when the database owner revokes all the user's privileges.

1.20.8.10 UTF8

Operating Condition

Syntax: SET UTF8 ON/OFF

Default: OFF

Controls the ability to use Unicode characters for string functions in applications and environments which will use higher character sets.

1.20.8.10 VERIFY

Operating Condition

Syntax: SET VERIFY COLUMN/ROW

Default: COLUMN

Mode: Multi-user

SET VERIFY, used in the multi-user environment, specifies the level of concurrency control as a row or a column within a row.

SET VERIFY allows you to specify whether R:BASE concurrency control will apply to individual columns within a row or to all columns in the row. When the level of concurrency control is set to COLUMN, R:BASE checks only the columns you change while you are editing. When the level of concurrency control is set to ROW, if you change data in any column, R:BASE checks every column in the row.

R:BASE concurrency control operates automatically when you are using a form in multi-user environments. Concurrency control includes autorefresh and verification. When you refresh or try to save a row, R:BASE checks whether data has been changed by another user and alerts you if it has changed. This prevents simultaneous changes that could corrupt the integrity of your data. The SET VERIFY command affects the operation of both autorefresh and verification when you are using a form.

When concurrency control is set to COLUMN, R:BASE looks for conflicts, those instances when two users have both modified the same column. When R:BASE detects a conflict in at least one field:

- R:BASE displays all of the other user's changes in the appropriate fields.
- Where there is no conflict, changes you made continue to be displayed.
- R:BASE displays a message informing you that data has changed.

When concurrency control is set to ROW, R:BASE looks for a change to any column in the whole row, whether it is a conflict or not. When R:BASE detects a change:

- R:BASE displays all of the other user's changes in the appropriate fields.
- Where there is no conflict, changes you made to the data are discarded.
- R:BASE displays a message informing you that data has changed.

Whether concurrency control is set to COLUMN or ROW, you can review the changes and then continue editing the data in the form. After autorefresh, R:BASE prompts you to press any key to continue editing. After verification, you can either move on to your next task or edit the data again. If you choose to move on when the level of concurrency control is set to COLUMN, you will be discarding any changes you made that are still displayed. R:BASE prompts you to press [Esc] if you want to move on, or to press [Enter] if you want to edit the displayed data.

When you edit data in a form, concurrency control is always enforced.

When you enter data in a form, concurrency control is enforced only when you are entering values in fields defined with a same-table look-up, or when you return to a row in a region that you had entered previously.

The first command line in the following example sets the level of concurrency control to check for changes in the entire row. The second command line starts an editing session using the form named *custform*.

```
SET VERIFY ROW
EDIT USING custform
```

1.20.8.10 WAIT

Operating Condition

Syntax: SET WAIT value

Range: 0 to 16383 seconds

Default: 4

Mode: Multi-user

SET WAIT sets the minimum number of seconds to retry the last requested resource (a table or database) before halting execution. Rather than aborting execution, SET WAIT allows you to set a length of time for R:BASE to keep trying to access a resource.

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

For commands that wait for resources, the precise period of the wait is at least as long as the time specified. On some computers, processing requirements may extend the length of the wait to longer than one second for each second designated.

When you enter a command from the R> prompt and the waiting period expires, R:BASE displays a message informing the user that the resource is unavailable. When the command runs as part of a command file, however, and the waiting period expires, R:BASE displays the unavailable resource message, ignores the command, and goes on to the next command.

The following command tells R:BASE to retry the last requested resource for approximately 20 seconds.

```
SET WAIT 20
```

You can also adjust the interval in which R:BASE tries the command during the SET WAIT period.

For more information, see [INTERVAL](#).

1.20.8.10 **WALKMENU**

Operating Condition

Syntax: SET WALKMENU ON/OFF

Default: OFF

Allows shortcut access to menus

SET WALKMENU is a menu shortcut function allowing the user to access menu selections by typing the beginning characters (up to when a match is made) of their names. Pressing any navigational keys (such as [Home] or [Page Up]) clears the buffer containing the keystrokes entered by the user while traversing the menu list. Any keystrokes not resulting in a match are not stored in the buffer, causing a beep.

Also see Walkmenu Time Interval

1.20.8.11 **WAREKI**

Operating Condition

Syntax: SET WAREKI ON/OFF

Default: OFF

Supports Japanese WAREKI style (year of the emperor) date formats. The setting is specific to R:BASE use for a computer whose system locale is configured for Japan.

1.20.8.11 **WHILEOPT**

Operating Condition

Syntax: SET WHILEOPT ON/OFF

Default: ON

SET WHILEOPT improves the optimization and processing of [WHILE ...ENDWHILE](#) loops within applications by pre-compiling variables used within the WHILE loop. Follow these guidelines:

- Don't clear your WHILE variable(s).
- Don't define variables within your WHILE loop, only outside the loop; values can change within the loop.
- Adhere to the syntax rules for the [SWITCH](#) statement by making sure that the argument for the SWITCH statement is an expression.
- If you issue multiple [SET VARIABLE](#) commands on a single command line, those variables will not be optimized. If you want to increase the speed for that loop, put those SET VARIABLE commands on separate lines.

The WHILEOPT setting must be changed in a command file. The setting cannot be saved to the configuration file.

1.20.8.11 **WIDTH**

Operating Condition

Syntax: SET WIDTH value

Range: 40 to 5000 characters

Default: 79

SET WIDTH controls the width of a data line that R:BASE directs to the printer, screen, or file when using the [BACKUP](#), [COMPUTE](#), [CROSSTAB](#), [DISPLAY](#), [LIST](#), [SELECT](#), [TYPE](#), [UNLOAD](#), or [WRITE](#) commands. Use the [NAMEWIDTH](#) setting to control the number of characters for the first column in the displayed list.

Do not set the width to a number greater than the number of characters your printer can fit on a line; a typical page and computer screen display 80 characters. WIDTH does not affect report generation; each report defines the width of a data line.

Note: The 5000-character maximum does not apply to the TYPE or DISPLAY commands, which continue to have 256-character width limits.

1.20.8.11 WINAUTH

Operating Condition

Syntax: SET WINAUTH ON/OFF

Default: OFF

SET WINAUTH controls if R:BASE will automatically authenticate database connections through Integrated Windows Authentication (IWA) with Windows Active Directory. When WINAUTH is set ON, the security features of the Windows clients and servers allows access to an R:BASE database when the Windows user name and password matches a defined R:BASE database user name and password. The current Windows user information on the client computer is supplied to R:BASE and does not prompt user for additional user name and password.

In order to use this feature, R:BASE database administrators must create an owner and user privileges for the database files to take full advantage of IWA support.

In addition to creating the database user privileges, the operating condition WINAUTH must be set to ON when prior to users connecting to the database. WINAUTH can be set within the R:BASE configuration file, or defined within an application startup file.

Requirements:

- WINAUTH must be set ON before connecting to the database.
- Database must contain an owner and user privileges that match Windows network accounts.

Notes:

- R:BASE will accept user names that contain spaces.
- Passwords are case-sensitive.
- If the authentication exchange initially fails to identify the user, R:BASE will prompt the user for a user name and password.
- A new CVAL Function (CVAL('WINAUTH')) has been established to check if WINAUTH is on.
- When an R:BASE user successfully connects to the database via IWA, the current R:BASE [USER](#) identifier is set to the current Windows user name.

1.20.8.11 WINBEEP

Operating Condition

Syntax: SET WINBEEP ON/OFF n

Default: OFF

Allows R:BASE to access a subset of the Sound Events in Windows. WINBEEP command will use the current system sound schema.

When set to OFF R:BASE will use a standard Windows sound for all errors and the [BEEP](#) command. When set to ON certain sounds, as set in the Windows Control Panel, will be used instead depending on which sound type is used. Below is a table of Types and their corresponding sound event. You must set the actual sounds used in the Windows Sounds Control Panel Applet. You will also need to ensure that your speakers are Un-Muted and working properly. In either case, WINBEEP ON or WINBEEP OFF, if you have the "No Sounds" scheme selected in your Sounds Control Panel you will not hear any beeps from R:BASE for Windows.

Sound Type	Sound Event
0	SYSTEM
1	SYSTEMSTART
2	SYSTEMEXIT
3	SYSTEMHAND
4	SYSTEMASTERISK
5	SYSTEMQUESTION
6	SYSTEMEXCLAMATION
7	SYSTEMWELCOME
8	SYSTEMDEFAULT

Example

```

SET CAPTION ' '
SET VAR VRows INTEGER = 0
SET VAR VMsg TEXT = NULL
SELECT COUNT(*) INTO VRows INDIC IVRows FROM TableName
IF VRows = 0 THEN
    CLS
    SET WINBEEP ON 2
    BEEP
    PAUSE 2 USING 'No Record(s) on File!' AT CENTER CENTER
    SET WINBEEP OFF
    GOTO Done
ELSE
    CLS
    SET WINBEEP ON 1
    BEEP
    SET VAR VMsg = ((CTXT(.VRows)) & 'Record(s) on File!')
    PAUSE 2 USING .VMsg AT CENTER CENTER
    SET WINBEEP OFF
    GOTO Done
ENDIF

LABEL Done
CLS
CLEAR ALL VAR

```

1.20.8.11!WRAP

Operating Condition

Syntax: SET WRAP ON/OFF

Default: ON

Text in fields that have NOTE and TEXT data types will wrap in forms and [FILLIN](#) windows, variables, and reports.

1.20.8.11WRITECHK

Operating Condition

Syntax: SET WRITECHK ON/OFF

Default: OFF

SET WRITECHK ON tells R:BASE to verify every write to disk.

1.20.8.11ZERO

Operating Condition

Syntax: SET ZERO ON/OFF

Default: OFF

SET ZERO allows a null to be treated as a zero in a mathematical expression involving INTEGER, NUMERIC, REAL, DOUBLE, CURRENCY, DATE, DATETIME, or TIME data types. With ZERO set on, R:BASE returns a negative number when you subtract an integer from a null. With ZERO set off, the same computation results in a null. R:BASE stores the setting with the database.

1.20.8.11ZOOMEDIT

Operating Condition (**R:BASE for DOS only**)

Syntax: SET ZOOMEDIT ON/OFF

Default: OFF

Toggles the field expansion method.

Set ZOOMEDIT to on if you want to open REdit, the R:BASE text editor, when you zoom in on a NOTE filed in a form. When ZOOMEDIT is off, a dialog box that wraps text is opened instead.

1.20.9 SET STATICVAR

Use the SET STATICVAR command to define or redefine a static variable value and/or data type.

```
SET STATICVAR varname datatype
```

```
SET STATICVAR varname [datatype] = [
    .varname
    &varname
    value
    <expression>
    NULL
]
```

```
SET STATICVAR varname [datatype] = colname IN tblview [WHERE clause]
```

Options**colname**

Specifies a column name. The column name is limited to 128 characters. In a command, you can enter #c, where #c is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (tblname.colname).

datatype

Specifies an R:BASE data type for the static variable. See Data Types.

(expression)

Determines a value using a text or arithmetic formula. The expression can include constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

IN tblview

Specifies a table or view.

NULL

Sets the static variable equal to NULL.

value

Sets the static variable equal to a specified value. A value is a constant amount, text string, date, or time, or the value assigned to *varname*.

varname

Specifies a static variable name, which must be unique among both the static variable names within the database and the global variable names for the R:BASE session. The static variable name is limited to 128 characters.

&varname

Sets the first variable equal to the exact contents of a second variable; the ampersand tells R:BASE to evaluate the contents of the variable first.

For example, if *varname* is the text string (2+3), then *&varname* is the value 5.

.varname

Sets the first variable equal to the exact contents of a second variable.

For example, if *varname* is the text string (2+3), then *.varname* is (2+3).

WHERE clause

Limits rows of data. For more information, see the [WHERE](#) Clause.

About the SET STATICVAR Command

The SET STATICVAR command defines static variables, which are part of a connected database. Once defined, the variables are created every time the database is connected. When a database is disconnected, the static variables associated with the database are cleared. Static variable can be used just like normal variables with [SET VARIABLE](#) commands and in expressions. The basic syntax is just like the SET VARIABLE command except you use the word STATICVAR instead of VARIABLE (or VAR).

The [UNLOAD](#) command will create SET STATICVAR commands for the static variables defined for the connected database.

To display the current static variables use the [SHOW STATICVAR](#) command. To clear one or more static variables use the [CLEAR](#) command. CLEAR ALL VAR will clear all variables (global and static), but immediately recreates the static variables, just like R:BASE does with system variables.

Important: It is recommended to define each static variable as a [separate](#) entry in the database. Defining multiple static variables with a single SET STATICVAR command should be avoided, as the multiple variables are defined as a group and are retained as a single database entry. If the variable group are always used as a set, then they are more ideal to manage. If the intent is to store each static variable as a separate item, each variable must be defined with a separate SET STATICVAR command.

The benefits of static variables include:

- Database unloads will no longer trip over missing global variables
- Users would not have to remember to define required variables used in views
- With CVAL functions, the current computer and environment may be easily captured
- Distributed applications could contain static variables specific to each customer

Static variable have the following restrictions:

- The variable name is not an R:BASE reserved word.
- The variable name begins with a letter, contains only letters, numbers, and the following special characters: #, \$, _, and %.
- The variable definitions are limited to 512 characters.

It is good programming practice to always define the data type for the variable before assigning it a value. When defining a variable as a text string, enclose the text string in single quote marks (or the current QUOTES character); otherwise, it might be interpreted as an arithmetic expression.

Assigning a Data Type to a Static Variable

The *datatype* option refers to one of the valid R:BASE data types. You can define a static variable to have a NOTE data type, but R:BASE treats it as TEXT for most uses. You can also specify the precision and scale for NUMERIC data types.

The *datatype* option creates a static variable, determines its data type, and sets its value to null. Use this option to define a static variable's data type before assigning a value to the variable.

For an existing static variable, you can use the *datatype* option to change the data type, but it is recommended to use one of the conversion functions. If you change the data type, the new data type must be compatible with the current variable value; if the variable is not compatible, R:BASE displays an error message and leaves the value and data type unchanged. If you change a variable with a TEXT data type to a non-compatible data type, R:BASE changes the value to null.

Assigning a Value to a Static Variable

The *value* option is a data value or constant, such as 10, TOM, 3.1416, or \$17.23. If the static variable already exists, any new value must have a data type that is compatible with that variable. If the static variable does not exist, R:BASE defines the variable's data type. You can also define the variable's data type in this command before assigning it a value.

Setting the Value of a Static Variable to Another Variable

When you set a static variable to the value of another variable, the second variable must be a dot variable (.) or an ampersand (&) variable.

When you precede a variable with a dot (.), R:BASE uses the value stored in the variable as if it were a constant.

When you precede a variable with an ampersand (&), R:BASE first evaluates the value contained in the ampersand variable. For example, consider the following uses of the command:

```
SET STATICVAR vM TEXT = 'Multi'
SET STATICVAR vP TEXT = 'Purpose'
SET STATICVAR vMP TEXT = '(vM + vP)'
SET STATICVAR vMPValue = .vMP
SET STATICVAR vMPCompute = &vMP
```

When the third command line runs, the variable *vMP* will contain (vM + vP). When the fourth command line runs, variable *vMPValue* will also contain (vM + vP) because the dot tells R:BASE to set the value as an exact match to the contents of variable *vMP*. When the fifth command line runs, variable *vMPCompute* will contain MultiPurpose (the concatenation of Multi and Purpose) because the ampersand tells R:BASE to compute the contents of variable *vMP*.

As shown in the example above, an ampersand variable can contain one command or part of one command, such as an expression. The first variable is set to the computed value of the ampersand variable. Below is an example:

```
1. SET STATICVAR v1 TEXT
2. SET STATICVAR v2 INTEGER
3. SET STATICVAR v1 = '((50 + 100) / 2)'
4. SET STATICVAR v2 = &v1
5. SHOW STATICVAR
```

1. Sets the data type for the variable *v1* to TEXT.
2. Sets the data type for the variable *v2* to INTEGER.
3. Sets variable *v1* to a text value that is a valid arithmetic expression.
4. Sets variable *v2* to &*v1*.
5. Displays the value of all static variables.

R:BASE computes the expression contained in *v1* and assigns the calculated value to *v2*. When R:BASE sees a variable name preceded by ampersand, it treats the contents of the variable as if it was entered from the keyboard. The [SHOW STATICVAR](#) display would like the following:

Variable	= Value	Type
-----	-----	-----
<i>v1</i>	= ((50 + 100) / 2)	TEXT
<i>v2</i>	= 75	INTEGER

Setting a Static Variable to an Expression

An (*expression*) can be either an arithmetic operation that combines two or more items in an arithmetic computation, or a string expression that concatenates two or more text items, or uses a TEXT function. The items can be values or the values contained in variables.

If you do not predefine the data type of a static variable, the original data type of each item determines the data type of the result. For example, if you add a variable that has an INTEGER data type to a variable that has a REAL data type, the resulting variable has a REAL data type unless you define the result to be an INTEGER data type.

If any item in an arithmetic expression is null, the result will be null unless you specify [SET ZERO ON](#).

Assigning Column Values in a Table or View

If you specify a table or view in a SET STATICVAR command, you can include an optional WHERE clause to indicate which row to use. If you do not include the WHERE clause, R:BASE uses the value for the column in the first row.

You must have [SELECT privileges](#) on the table to use this form of SET STATICVAR.

Examples

Defines the *vMessage* static variable to have a TEXT data type.

```
SET STATICVAR vMessage TEXT
```

Defines the *vReal* static variable to have a REAL data type, and assigns it the value 100.9.

```
SET STATICVAR vReal REAL = 100.9
```

Defines the *vNum* static variable to have a NUMERIC data type having a precision of 9 and scale of 3.

```
SET STATICVAR vNum NUMERIC (9,3)
```

Assigns the integer value 14322 to the *vNum* static variable.

```
SET STATICVAR vNum = 14322
```

Assigns the value of the above *vNum* variable to the *vTwo* static variable.

```
SET STATICVAR vTwo = .vNum
```

Assigns the value 03/25/2022 to the *vLtrDate* static variable.

```
SET STATICVAR vLtrDate = ('12/25/2021' + 90)
```

Assigns the *vCompanyName* static variable from the row in the *LicenseInformation* table where the *LicenseNumber* value is equal to BM-68215.

```
SET STATICVAR vCompanyName TEXT = CompanyName IN LicenseInformation WHERE
LicenseNumber = 'BM-68215'
```

1.20.10 SET VARIABLE

Use the SET VARIABLE command to define or redefine a variable value and/or data type.

```

SET VARIABLE [ MDI ] varname datatype

SET VARIABLE [ MDI ] varname [ datatype ] = [ .varname
&varname
value
(expression)
NULL ]

SET VARIABLE [ MDI ] varname [ datatype ] = colname IN tblview [ WHERE clause ]

```

Options

'
Indicates that this part of the command is repeatable.

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter *#c*, where *#c* is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (*tblname.colname*).

datatype

Specifies an R:BASE data type for the variable. See Data Types.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

IN tblview

Specifies a table or view.

MDI

Specifies to define a variable local to an MDI form.

NULL

Sets the variable equal to NULL.

value

Sets the variable equal to a specified value. A value is a constant amount, text string, date, or time, or the value assigned to *varname*.

varname

Specifies a variable name, which must be unique among the variable names within the database. The variable name is limited to 128 characters.

&varname

Sets the first variable equal to the exact contents of a second variable; the ampersand tells R:BASE to evaluate the contents of the variable first.

For example, if *varname* is the text string (2+3), then *&varname* is the value 5.

.varname

Sets the first variable equal to the exact contents of a second variable.

For example, if *varname* is the text string (2+3), then *.varname* is (2+3).

WHERE clause

Limits rows of data. For more information, see the [WHERE](#) Clause.

About the SET VARIABLE Command

Variables identify a changeable value. R:BASE provides four kinds of variables: global, [static](#), error, and system. The SET VARIABLE command defines global variables, which are temporary variables that exist within R:BASE, but are not part of any database. Global variables remain in memory until you clear them or exit from R:BASE. R:BASE sets error and system variables internally.

Global variables have several uses: they can provide a temporary value in a command, hold the result of a calculation, act as a counter, or capture keyboard entries for use with menus or screens. The most common method of defining variables is to assign the variable value with the SET VARIABLE command.

Variable names have the following restrictions:

- The variable name is not an R:BASE reserved word.
- The variable name begins with a letter, contains only letters, numbers, and the following special characters: #, \$, _ , and %.

It is good programming practice to always define the data type for the variable before assigning it a value, unless you are setting a variable to a column value or using the variable in the [CHOOSE](#) command.

When defining a variable as a text string, enclose the text string in single quote marks (or the current QUOTES character); otherwise, it might be interpreted as an arithmetic expression.

Assigning a Data Type to a Variable

The *datatype* option refers to one of the valid R:BASE data types. You can define a variable to have a NOTE data type, but R:BASE treats it as TEXT for most uses. You can also specify the precision and scale for NUMERIC data types.

The *datatype* option creates a variable, determines its data type, and sets its value to null. Use this option to define a variable's data type before assigning a value to the variable. To set multiple variables in the same command, separate the variables by a comma or the current delimiter.

For an existing variable, you can use the *datatype* option to change the data type, but it is recommended to use one of the conversion functions. If you change the data type, the new data type must be compatible with the current variable value; if the variable is not compatible, R:BASE displays an error message and leaves the value and data type unchanged. If you change a variable with a TEXT data type to a non-compatible data type, R:BASE changes the value to null.

Assigning a Value to a Variable

The *value* option is a data value or constant, such as 10, TOM, 3.1416, or \$17.23. If the variable already exists, any new value must have a data type that is compatible with that variable. If the variable does not exist, R:BASE defines the variable's data type.

You can also define the variable's data type in this command before assigning it a value.

Setting the Value of a Variable to Another Variable

When you set a variable to the value of another variable, the second variable must be a dot variable (.) or an ampersand (&) variable.

When you precede a variable with a dot (.), R:BASE uses the value stored in the variable as if it were a constant.

When you precede a variable with an ampersand (&), R:BASE first evaluates the value contained in the ampersand variable. For example, consider the following uses of the command:

```
SET VARIABLE vM TEXT = 'Multi'
SET VARIABLE vP TEXT = 'Purpose'
```

```
SET VARIABLE vMP TEXT = '(vM + vP)'
SET VARIABLE vMPValue = .vMP
SET VARIABLE vMPCompute = &vMP
```

When the third command line runs, the variable *vMP* will contain (vM + vP). When the forth command line runs, variable *vMPValue* will also contain (vM + vP) because the dot tells R:BASE to set the value as an exact match to the contents of variable *vMP*. When the fifth command line runs, variable *vMPCompute* will contain MultiPurpose (the concatenation of Multi and Purpose) because the ampersand tells R:BASE to compute the contents of variable *vMP*.

As shown in the example above, an ampersand variable can contain one command or part of one command, such as an expression. The first variable is set to the computed value of the ampersand variable. Below is an example:

```
1. SET VARIABLE v1 TEXT, v2 INTEGER
2. SET VARIABLE v1 = '((50 + 100)/ 2)'
3. SET VARIABLE v2 = &v1
4. SHOW VARIABLE
```

1. Sets the data types for variables *v1* and *v2* to TEXT and INTEGER, respectively.
2. Sets variable *v1* to a text value that is a valid arithmetic expression.
3. Sets variable *v2* to &*v1*.
4. Displays the value of all variables, including the system variables.

R:BASE computes the expression contained in *v1* and assigns the calculated value to *v2*. When R:BASE sees a variable name preceded by ampersand, it treats the contents of the variable as if it was entered from the keyboard. The [SHOW VARIABLE](#) display would like the following:

Variable	= Value	Type
-----	-----	-----
#DATE	= 12/23/2022	DATE
#TIME	= 02:29 PM	TIME
#PI	= 3.14159265358979	DOUBLE
SQLCODE	= 0	INTEGER
SQLSTATE	= 00000	TEXT
#NOW	= 12/23/2022 02:29 PM	DATETIME
v1	= ((50 + 100)/ 2)	TEXT
v2	= 75	INTEGER

Setting a Variable to an Expression

An (*expression*) can be either an arithmetic operation that combines two or more items in an arithmetic computation, or a string expression that concatenates two or more text items, or uses a TEXT function. The items can be values or the values contained in variables.

If you do not predefine the data type of a variable, the original data type of each item determines the data type of the result. For example, if you add a variable that has an INTEGER data type to a variable that has a REAL data type, the resulting variable has a REAL data type unless you define the result to be an INTEGER data type.

If any item in an arithmetic expression is null, the result will be null unless you specify [SET ZERO ON](#).

Assigning Column Values in a Table or View

If you specify a table or view in a SET VARIABLE command, you can include an optional WHERE clause to indicate which row to use. If you do not include the WHERE clause, R:BASE uses the value for the column in the first row.

You must have [SELECT privileges](#) on the table to use this form of SET VARIABLE.

In instances where you're building a dynamic SET VARIABLE command based on previous options made, you must use an ampersand variable in place of a column or table name, for example:

```
CHOOSE vtab FROM #TABLES
CHOOSE vcol FROM #COLUMNS IN &vtab
SET VARIABLE vnewpr = &vcol IN &vtab
```

Enter the table and column names into the *vtab* and *vcol* variables first. You can do this by using the CHOOSE #TABLES and CHOOSE #COLUMNS commands, as shown in the above example. The [CHOOSE](#) command displays a menu of tables or columns from which to choose. By using ampersand variables to hold the table and column names, you can use the same SET VARIABLE command to get values from different columns in a table or from different tables. Each time SET VARIABLE requests a column, it retrieves information from the first row in the table stored in &vtab.

NOTE: You can define multiple variables with a single SET VARIABLE command when you set the value of the variables to the value of columns in a table. However, when capturing column data into variables, it is better to use the SELECT command; specifically, [SELECT INTO](#). SELECT INTO is the SQL compliant command when capturing table data into variables.

Using Local Variables within MDI Forms

The MDI keyword creates variables used in a MDI form, where with multiple instances of the MDI form, each form will have separate variables. Expressions that reference that variable will simply use the variable name. If a global variable is defined with the same name as a MDI variable, R:BASE will find the global variable first.

The link to which MDI form to use is based on the value of the RBTI_FORM_ALIAS variable, which holds the alias name. Different instances of MDI forms need different alias names.

When defining local variables to a MDI form, the "On Before Start" form EEP cannot be used for the SET VAR MDI ... command, as the form must be initialized first.

Examples

Defines the vText variable to have a TEXT data type.

```
SET VARIABLE vtext TEXT
```

Defines the vReal variable to have a REAL data type, and assigns it the value 100.9.

```
SET VARIABLE vReal REAL = 100.9
```

Defines the vNumer variable to have a NUMERIC data type having a precision of 9 and scale of 3.

```
SET VARIABLE vNumer NUMERIC (9,3)
```

Assigns the integer value 14322 to the vNum variable.

```
SET VARIABLE vNum = 14322
```

Assigns the value of the above vNum variable to the vTwo variable.

```
SET VARIABLE vTwo = .vNum
```

Assigns the value 03/25/2022 to the vLtrDate variable.

```
SET VARIABLE vLtrDate = ('12/25/21' + 90)
```

Assigns to the vFullName variable the value of the full name formed by concatenating the values in the vFirstName and vLastName variables. The ampersand inserts a space between the two values.

```
SET VARIABLE vFullName = (.vFirstName & .vLastName)
```

Defines the vCounter INTEGER variable for a MDI form.

```
SET VARIABLE MDI vCounter INTEGER = 0
```

Assigns the first name, last name, and phone number to the variables from the row in the Contact where ContID is equal to 1001.

```
SET VARIABLE vContFName = ContFName, vContLName = ContLName, vContPhone = ContPhone IN
Contact WHERE ContID = 1001
```

OR SQL compliant variation (See [SELECT INTO](#)):

```
SELECT ContFName, ContLName, ContPhone INTO vContFName INDI iv1, vContLName INDI iv2,  
vContPhone INDI iv3 FROM Contact WHERE ContID = 1001
```

1.20.11 SETFOCUS

Use the SETFOCUS command to return focus to an MDI form from within an application.

SETFOCUS alias

Options

alias

Specifies a window name, which is the name given to the instance of a form started with the MDI option when using the AS *alias* option with the [BROWSE USING](#), [EDIT USING](#) or [ENTER](#) command.

1.20.12 SHARECPY

Use the SHARECPY command to copy files, including files being shared on a network.

SHARECPY filesource filetarget

Options

filesource

Specifies the file to be copied, the source file

filetarget

Specifies the file to receive the copy, the target file

About the SHARECPY Command

The command is particularly useful for making a backup copy of a database that is shared on a network.

Although you should **not attempt to copy the database while users are actively making changes to it**, the command allows you to copy it without having to make sure that all users have disconnected from the database.

When SHARECPY is in the process of copying files, and encounters locked bytes in a file (which means it cannot read those bytes), the resulting file will have those bytes replaced with zeros. Such a resulting file will most likely be corrupted, but the command will read what it can.

When you copy files from one drive to another, include the drive letter of the drive you want to copy to.

When you copy files from one directory to another directory, the name of the directory you want to copy to must be different from the current directory.

Naming Copied Files

When you copy a file from one drive or directory to another drive or directory, you can keep the same name for the new file, or you can give the file a new name. If you want to make a copy of a file on the same directory, you must give the file a different name. If the new name is the name of an existing file, SHARECPY replaces the existing file with the file you are copying.

Using Wildcards with the SHARECPY Command

When copying a group of files, you can use the operating-system wildcards (?) or (*) in the file specification. The settings for the R:BASE special characters SINGLE and MANY do not affect the operating system wildcards.

When the target name is the same length as the source name, the command succeeds. However, if the target name is shorter or longer than the source name, the results are unpredictable.

Backing Up with the SHARECPY Command

You can use the SHARECPY command to backup your database, command files, and programs. When copying a database, copy all four database files as a set.

If you change the file extension when you use SHARECPY to backup your database files, you must change the extension back to the original extension to use the database in R:BASE. The SHARECPY command does not copy files onto multiple disks. If your database is too large to fit on one disk, use the [BACKUP](#) or [UNLOAD](#) commands.

Suggestion

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

Examples

The following copies the copies a database called mydb from the current directory to another directory called BACKUP:

```
SHARECPY mydb.rb? C:\backup\*.*
```

The following command copies any files in the DBS directory that begin with MYDATA and have the extension .RX? to the ADMIN directory on the current drive using the same file names. The files with the .RX? extension could be the four disk files that contain an R:BASE database: MyData.RX1, MyData.RX2, MyData.RX3, and MyData.RX4.

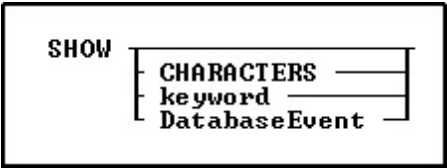
```
SHARECPY /DBS/MyData.rx? /admin
```

The following command copies the MYDATA database files with an .RX? extension and changes the extension to .SV?

```
SHARECPY MyData.rx? MyData.sv?
```

1.20.13 SHOW

Use the SHOW command to display the status of system parameters, format conditions, default conditions, and database events.



Options

CHARACTERS

Shows the current settings for the BLANK, DELIMIT, LINEEND, MANY, PLUS, QUOTES, SEMI, and SINGLE special characters.

DatabaseEvent

Shows the command file assigned to the specified database event. Database events can include the following:

- OnError

- OnBeforePack
- OnPack
- OnBeforeConnect
- OnConnect
- OnFirstConnect
- OnBeforeDisconnect
- OnAfterDisconnect
- OnExit

keyword

Shows the current setting for the environment condition specified in the *keyword* option.

About the SHOW Command

Use any of the [SET](#) keywords in a SHOW command to display an environment setting. You can also display special characters and environment settings by selecting Settings > Configuration Settings from the main menu bar.

The following commands are not displayed in a standard SHOW and must be requested manually.

- [AUTOCONVERT](#)
- [AUTODROP](#)
- [AUTORECOVER](#)
- [AUTOROWVER](#)
- [AUTOSYNC](#)
- [AUTOUPGRADE](#)
- [COMPATIBLE](#)
- [FASTFK](#)
- [FASTLOCK](#)
- [PAGELOCK](#)
- [PAGEMODE](#)
- [PASSTHRO](#)
- [QUALCOLS](#)
- [SERVER](#)
- [WHILEOPT](#)

Examples

The following command displays all characters and settings.

```
SHOW
```

The following command displays settings for the special characters BLANK, DELIMIT, LINEEND, MANY, PLUS, QUOTES, SEMI, and SINGLE.

```
SHOW CHARACTERS
```

The following command displays the status of rules checking.

```
SHOW RULES
```

The following command displays the current user identifier.

```
SHOW USER
```

The following command displays the front end build of the product.

```
SHOW BUILD
```

The following command displays the command file assigned to the "On Connect" database event.

```
SHOW OnConnect
```

What follows is the output from the SHOW command:

```
BLANK
DELIMIT      ,
LINEEND      ^
SEMI         ;
QUOTES       '
PLUS         +
SINGLE        _
MANY         %
IDQUOTES     `
```

```
DATE format      MM/DD/YYYY
DATE sequence    MMDDYY
Century threshold YEAR is 30
Default CENTURY is 19
TIME format      HH:MM AP
TIME sequence    HHMM
LINES/page      20
WIDTH/line      79
TOLERANCE              0.
NULL symbol        -0-
Lock WAIT time     4
Lock INTERVAL time (in .1 sec)      5
REFRESH TIME       0
MAXTRANS           201
EDIT verification level COLUMN
FILES              25
PRINTER
NOTE column PADding percentage 10
CURRENCY SYMBOL    '$'
CURRENCY LOCATION  PREF
CURRENCY DIGITS    2
CURRENCY CONVENTION B
```

```
(AND          ) ON  Use SQL precedence for ANDs and ORs
(ANSI         ) ON  ANSI SQL name restrictions
(ANSIOutput) OFF Output files will be created as ANSI files without specifying ANSI
keyword
(AUTOCOMM     ) OFF AUTOMatically COMMIT after each command
(AUTOSKIP     ) OFF AUTOMatically SKIP to the next field when editing
(BELL         ) ON  Sound the BELL on an error
(BOOLEAN      ) OFF BOOLEAN constants in expressions will be treated as type BOOLEAN
values
(CASE         ) OFF Distinguish between UPPER and lower CASE
(CACHEDATA) OFF Enable file 2 and 3 caching between commands and connects
(CHECKPROP) OFF Display PROPERTY/GETPROPERTY ERROR messages during processing
(CLEAR        ) ON  CLEAR data buffers after modifications
(CMDHIST      ) ON  Update R> command history when running commands from a dialog
(CMPAUSE      ) OFF Cascade Modal PAUSE operations
(COLCHECK     ) ON  COLUMNS will be CHECKed for consistency when making views
(DEBUG        ) OFF Debug command is effective
(ECHO         ) OFF ECHO input from command files
(EOFCHAR      ) ON  Add EOF CHARACTER to the end of output files
```

(EQNULL) OFF Treat null EQUAL NULL as a match
 (ERROR) ON Display ERROR messages during processing
 (ESCAPE) ON ESCAPE allowed to abort processing

 (EXPLODE) OFF Exploding windows on dialog boxes
 (FEEDBACK) OFF Display FEEDBACK status during command processing
 (HEADINGS) ON Display column HEADINGS on SELECT and TALLY commands
 (INSERT) ON Use INSERT key as a toggle
 (LAYOUT) OFF Save the LAYOUT of data displayed on Browse/Edit
 (MANOPT) OFF Disable optimizer table reordering
 (MESSAGES) ON Display informational MESSAGES during processing
 (MIRROR) OFF Maintains MIRROR images of database files
 (MULTI) ON MULTI-user database access capability
 (NOCALC) OFF Generate NOCALC commands when doing UNLOAD of data

 (NUMROUND) OFF Round any excess scale digits for numeric data types
 (ONELINE) OFF Display only one line per row in select output
 (PARSESEMI) OFF Enable Parsing of Semi Colons in ampersand variable processing
 (PROGRESS) OFF Display PROGRESS status during command processing
 (RADMIN) OFF Network database administration
 (RECYCLE) OFF RECYCLE unused blocks in the RX2 data file
 (REVERSE) ON REVERSE video highlighting on data entry/edit
 (ROWLOCKS) ON ROW LOCK database access capability
 (RULES) ON Check data validation RULES
 (SCRATCH) C:\Users\JOSEMA~1\AppData\Local\Temp --SCRATCH files location
 (SHORTNAME) OFF DIR command option. Show short file names or not.
 (SORT) OFF Optimize SORTing to minimize disk space used
 (SORTMENU) ON Sort menu items for selected menu types
 (STATICDB) OFF STATIC DataBase schema mode
 (TRACE) OFF Interactive command file debugger
 (TRANSACTION) OFF Connect in TRANSACTION mode
 (UINOTIF) ON Front-End User Interface Notifications
 (UTF8) OFF UTF-8 characters supported in string functions
 (WALKMENU) OFF Search menus based on multiple keystrokes
 (WINAUTH) OFF Windows Authentication
 (WINBEEP) OFF Use selected WINDOWS BEEPs for beep sound
 (WRAP) ON Word WRAP text for display and edit
 (WRITECHK) OFF Verify each WRITE to disk
 (ZERO) OFF Treat nulls as ZERO in numeric calculations
 (ZOOMEDIT) ON Use rbEDIT for ZOOM on notes in forms

1.20.14 SHOW BUILD

SHOW BUILD displays the build number of the R:BASE executable (front-end).

SHOW BUILD

The build can also be captured, using the (CVAL('BUILD')) function.

See also:

(CVAL('VERSION'))
 (CVAL('VERSION BUILD'))

1.20.15 SHOW ERROR

Use the SHOW ERROR command without a variable name to display error-message status, which can be on or off.

```
SHOW ERROR [varname [AT scrnrow,scrncol]]
```

Options

AT scrnrow, scrncol

Defines the screen row and column at which to display the first character of the error message. Screen rows are between 1 and 25 and screen columns are between 1 and 80. This option is for R:BASE for DOS only.

varname

Specifies the name of the error variable, defined with the [SET ERROR VARIABLE](#) command. Alternatively, *varname* can be a global variable that captures the value of an error code from the defined error variable. If you enter the SHOW ERROR command without the *varname* option, R:BASE tells you whether error messages are on or off.

About the SHOW ERROR Command

Use SHOW ERROR with the name of an error variable to display the error message of the last R:BASE error that occurred.

The SHOW ERROR command with a variable name displays the error message of the error variable captured by the SET ERROR VARIABLE command. The variable can be either a currently defined error variable (defined with the SET ERROR VARIABLE command) or a global variable used to capture the value of the error variable.

You can capture an error-variable value in a global variable so that any additional commands that are processed do not change the value of the variable. Once you have captured the error number in a global variable, you can use the SHOW ERROR command to display the error message by referencing the global variable name.

Examples

The following series of commands:

- Define an error variable named *verrvar*.
- Connect to the *dbname* database.
- Capture the value of the *verrvar* error variable in the *vglobal* global variable.
- Check the value of *vglobal* and, if it is not 0 (zero), display the error message on screen line 20, beginning in screen column 10.

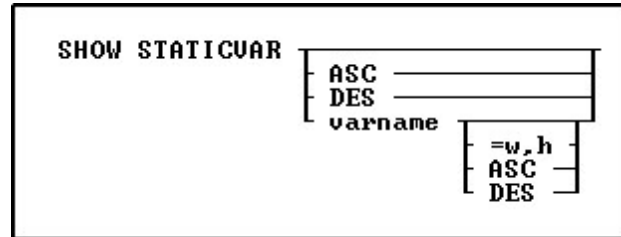
```
SET ERROR VARIABLE verrvar
CONNECT dbname
SET VARIABLE vglobal = .verrvar
IF vglobal <> 0 THEN
  SHOW ERROR vglobal AT 20 10
ENDIF
```

Used without a variable name, the following command displays the error-message status.

```
SHOW ERROR
(ERROR) ON Display ERROR messages during processing
```

1.20.16 SHOW STATICVAR

Use the SHOW STATICVAR command to display the current values and data types of static variables. Static variables are created with the [SET STATICVAR](#) command.



Options

ASC

Specifies to sort the variable output alphabetically in ascending order

DES

Specifies to sort the variable output alphabetically in descending order

varname

Specifies a static variable to be displayed. You can use wildcards to display variables that match a specific pattern.

=w,h

Specifies a maximum width and number of lines for the display of the variable. Long text variables are displayed on multiple lines. The =w,h option is limited to 30 characters and one line for these data types: CURRENCY, DATE, DATETIME, DOUBLE, INTEGER, NUMERIC, REAL, and TIME. When using the =w,h option for BIT, BITNOTE, LONG VARCHAR, LONG VARBIT, NOTE, TEXT, VARCHAR, and VARBIT variables, you can widen the display and set the total number of lines to display. If h is omitted, the value defaults to the number of lines needed to display the text for LONG VARCHAR, NOTE, TEXT, or VARCHAR data types, and only displays the file type for BIT, BITNOTE, LONG VARBIT, and VARBIT data types. Wildcards cannot be specified as part of the variable name with the width and line number parameters.

About the SHOW STATICVAR Command

Use SHOW STATICVAR to include static variable values in screen displays in a procedure, a command file, or at the R> Prompt. After displaying the variable values, R:BASE moves the cursor to the beginning of the next row.

You can also use SHOW STATICVAR or the [PAUSE](#) command to help debug command files by tracing the changing values of one or more variables.

Examples

The following command displays all defined static variables.

```
SHOW STATICVAR
```

The following displays all static variables in descending order.

```
SHOW STATICVAR DES
```

The following command displays the contents of the vMessage variable. If vMessage has a TEXT data type with a length of 50, R:BASE displays the first 25 characters on the first line, and the remaining 25 characters on the second line.. The setting for [SET WRAP](#) affects the display of variables.

```
SHOW STATICVAR vMessage =25
```

The following command displays all static variables that begin with the characters "Emp." The wildcard characters defined by the [SET SINGLE](#) and [SET MANY](#) settings must be used.

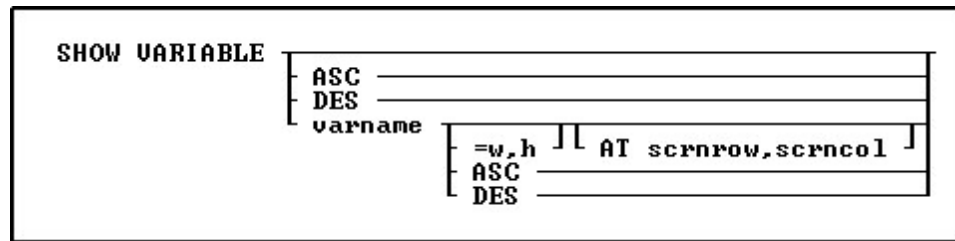
```
SHOW STATICVAR Emp%
```

The following command sorts all static variables beginning with "vCust" in ascending order. The wildcard characters defined by the SET SINGLE and SET MANY settings must be used.

```
SHOW STATICVAR vCust% ASC
```

1.20.17 SHOW VARIABLE

Use the SHOW VARIABLE command to display the current values and data types of global variables.



Options

ASC

Specifies to sort the variable output alphabetically in ascending order

AT scrnrow, scrncol

Defines the screen row and column where the first character of the value of the variable is displayed. Text values are left justified and numeric values are right justified within the display width specified by the =w option. Screen rows range from 1 to 25 and screen columns from 1 to 80. This option is for R:BASE for DOS only.

DES

Specifies to sort the variable output alphabetically in descending order

varname

Specifies a variable to be displayed. You can use wildcards to display variables that match a specific pattern.

=w,h

Specifies a maximum width and number of lines for the display of the variable. Long text variables are displayed on multiple lines. The =w,h option is limited to 30 characters and one line for these data types: CURRENCY, DATE, DATETIME, DOUBLE, INTEGER, NUMERIC, REAL, and TIME. When using the =w,h option for BIT, BITNOTE, LONG VARCHAR, LONG VARBIT, NOTE, TEXT, VARCHAR, and VARBIT variables, you can widen the display and set the total number of lines to display. If h is omitted, the value defaults to the number of lines needed to display the text for LONG VARCHAR, NOTE, TEXT, or VARCHAR data types, and only displays the file type for BIT, BITNOTE, LONG VARBIT, and VARBIT data types. Wildcards can be specified as part of the variable name with the width and line number parameters. When using the =w option the headers for the variable name, value and type are not displayed.

About the SHOW VARIABLE Command

Use SHOW VARIABLE to include variable values in screen displays in a procedure, a command file, or at the R> Prompt. After displaying the variable values, R:BASE moves the cursor to the beginning of the next row.

You can also use SHOW VARIABLE or the [PAUSE](#) command to help debug command files by tracing the changing values of one or more variables.

Examples

The following command displays all global variables and the system variables *#date*, *#pi*, *#time*, and *sqlcode*.

```
SHOW VARIABLE
```

The following displays all variables in descending order.

```
SHOW VARIABLE DES
```

The following command displays the contents of the *vMessage* variable. If *vMessage* has a TEXT data type with a length of 50, R:BASE displays the first 25 characters on the first line, and the remaining 25 characters on the second line.. The setting for [SET WRAP](#) affects the display of variables.

```
SHOW VARIABLE vMessage =25
```

The following command displays the contents of the *vaddr2* variable starting at row 10 and in column 15. If *vaddr2* has a TEXT data type with a length of 30, R:BASE displays only the first 22 characters on line 10. R:BASE displays the remaining characters starting at row 11 in column 15. The setting for [SET WRAP](#) affects the display of variables.

```
SHOW VARIABLE vaddr2 =22 AT 10 15
```

The following command displays all global variables that begin with the characters "emp." The wildcard characters defined by the [SET SINGLE](#) and [SET MANY](#) settings must be used.

```
SHOW VARIABLE emp%
```

The following command sorts all variables beginning with "vCust" in ascending order. The wildcard characters defined by the SET SINGLE and SET MANY settings must be used.

```
SHOW VARIABLE vCust% ASC
```

1.20.18 SHOW VERSION

SHOW VERSION displays product name, version, and build number of the R:BASE program.

SHOW VERSION

The version can also be captured, using the (CVAL('VERSION')) function.

See also:

```
(CVAL('BUILD'))  
(CVAL('VERSION BUILD'))
```

1.20.19 SKIP

Use the SKIP command in an entry/exit procedure (EEP) to automatically skip fields in a form.

**SKIP {ⁿ
TO fldname**

Options

n

Specifies the number of fields you want to skip, where n is a positive or negative integer. Use positive numbers to move forward through the field order in the form, and negative numbers to move backward. When you establish a value for n , count only those fields that allow you to enter or edit data.

SKIP 0 returns you to the same field you were in.

TO fldname

Specifies the field to which you want to skip, either forward or backward through the field order, where *fldname* is the column or variable name associated with a field.

About the SKIP Command

When an EEP is run, SKIP moves backward or forward through the fields in a form. SKIP operates within a form like the [Tab] key—it does not move to fields other than those in the current row or current table. You can use SKIP only in an EEP within a form.

This command is useful when you want to skip over certain fields in the form, depending on the data entered. A variable must always be associated with a field. You can set up an EEP to check the value entered in the field's variable, and then, depending on the value, skip to another field.

Notes:

- SKIP TO Fldname or SKIP n when used in a Tab Form will ONLY be applicable to that Tab (e.g., page) of the Tab Control. Therefore, you CANNOT use SKIP command to switch from any field on one page of the Tab Control to another page of the Tab Control. A Tab (page) of the tab control is similar to a "container". Thus, the SKIP command will ONLY work with objects on the same page of the Tab Control.
- The SKIP n command will base the field numbers to skip to based upon the [Tab] / [Shift]+[Tab] movement of the cursor, not the mouse. If you want to ensure the focus will land on a specific field, whether the user presses keys on the keyboard or uses the mouse, you can use SKIP TO Fldname or the PROPERTY command.
- You may also use the [PROPERTY](#) command as such; **PROPERTY <ComponentID> SET_FOCUS 'TRUE'** to achieve the same functionality.

Examples

The following commands cause a form to skip three fields if the *vcashtype* variable contains any value.

```
IF vcashtype IS NOT NULL THEN
  SKIP 3
ENDIF
```

The following commands cause a form to skip to the field containing the *amount* column if the *vcashtype* variable contains any value.

```
IF vcashtype IS NOT NULL THEN
  SKIP TO amount
ENDIF
```

1.20.20 SSQL

Use the SSQL command to pass SQL commands through R:BASE to the foreign data source, where the commands will be interpreted and executed.

SSQL SQLcommand

Options

SQLcommand

Specifies the SQL command to send.

About the SSQL Command

SSQL can be used to send structure modification commands that alter the foreign database, which are subject to the access rights and restrictions placed on the user by the foreign database.

SSQL requires R:BASE to already be connected to the data source. With a few exceptions, you can use every SQL command supported by the connected data source, such as [UPDATE](#) and [DELETE](#).

To use commands that return data, such as the [SELECT](#) command, the command syntax is supported without the use of SSQL, but [PASSTHROUGH](#) must be set ON.

1.20.21 STDCALL

Use the STDCALL command to call to a function or procedure as an external function declaration. Once the function or procedure name is declared, the DLCALL function may be used.

```
STDCALL FUNCTION 'FunctionName' [ ALIAS 'AliasName' ] ...
```

```
... < PTR datatype [ <size> ] > : datatype [ <size> ]
```

```
STDCALL VOID 'FunctionOrProcedureName' [ ALIAS 'AliasName' ] < PTR datatype [ <size> ] >
```

Options**ALIAS AliasName**

Specifies an alias name for the function.

datatype

Specifies an R:BASE data type.

FUNCTION

Specifies that a function is declared.

FunctionName

Specifies the function name.

FunctionOrProcedureName

Specifies the function or procedure name.

PTR

Specifies a data type pointer.

(size)

Defines the length for the TEXT or VARCHAR data type. Defines the precision and scale of a column of the DECIMAL or NUMERIC data type, if not the default of precision 9 and scale 0 (9,0).

VOID

Specifies that the function does not return a value.

About the STDCALL Command

STDCALL is a declaration in the R:BASE environment to set a reference (access) to the function.

Calls from any DLL must be declared at least once in the R:BASE session from which they will be referenced. The STDCALL declaration should be in a startup location for the R:BASE application. There is no limit to the number of declarations, but only make declarations that the application will use. Calling conventions specify how arguments are passed to a function, and how return values are passed back out of a function.

For best results, TEXT and VARCHAR data should be passed with the PTR (pointer) attribute. Any VARCHAR data type larger than 32K as a parameter, **MUST** be passed with the PTR attribute. It is important that the SIZE parameter on TEXT and VARCHAR be specified to avoid creating excess buffer space that has to be created from the declaration. The VARCHAR data type as a Return Value is restricted to 32K, but any SIZE up to 256MB can be passed as a pointer to the R:BASE variable. Modification of the data passed as pointer must be on the data pointed to.

STDCALL and [CDECL](#) are available as external function declaration commands. There is no performance difference between the two commands. When using CDECL, the calling program pushes the parameters on the stack and clears the stack on return. When using STDCALL, the calling program pushes the parameters on the stack but the called program clears the stack before doing the return. Choosing one over the other would depend on what style the DLL code being called is using.

Notes:

- The (size) parameter applies to TEXT and VARCHAR data types
- Parameters in the declaration are in the "reverse" order from the actual function or procedure in the DLL
- Parameters can be 0 to *n* parameters of any R:BASE data type
- Function names are case sensitive in the STDCALL or [CDECL](#) command declaration **ONLY**. The case must match the casing used in the DLL.
- The case is insensitive when used in the DLDCALL function.

Considerations:

If the function is declared as follows, then *nn* is less than or equal to 256 megabytes.

```
STDCALL FUNCTION 'somefunction' ( PTR VARCHAR (nn)) : INTEGER
```

If the function is declared as follows, the buffer for VARCHAR will have default size equal to 256MB because the size is omitted. It is advised to **avoid** this declaration use, unless that is the actual size of the data to be passed.

```
STDCALL FUNCTION 'somefunction' ( PTR VARCHAR ) : INTEGER
```

If the function is declared as follows, the buffer for VARCHAR will have default size equal to 32 kilobytes because the size is omitted.

```
STDCALL FUNCTION 'somefunction' ( INTEGER ) : VARCHAR
```

If the function is declared as follows, *nn* bytes less than or equal to 32 kilobytes will be returned because the size is specified. When the size is specified, the data is passed as a parameter or as a return value will be truncated to the size value, if the data is greater than the size.

```
STDCALL FUNCTION 'somefunction' ( INTEGER ) : VARCHAR(nn)
```

Examples:

Function Declarations:

```
STDCALL FUNCTION 'Num2WrdSTD' ( PTR INTEGER, PTR TEXT, DOUBLE ) : INTEGER
```

```
STDCALL FUNCTION 'MultIntSTD' ( INTEGER ) : INTEGER
```

```
STDCALL FUNCTION 'LCaseByVALSTD' ( PTR TEXT ) : TEXT
```

Procedure Declarations:

```
STDCALL VOID 'LCaseByREFSTD' ( PTR TEXT )
```

```
STDCALL VOID 'FunctionOrProcedureThatHasNoReturnValue' ( PTR TEXT (SIZE))
```

Windows API Declaration:

```
STDCALL FUNCTION 'GetCurrentDirectoryA' ALIAS 'GetCurrentDir' ( PTR TEXT,  
INTEGER ) : INTEGER
```

1.20.22 SUBTRACT

Use the SUBTRACT command to form a new table by subtracting rows in one table from a second table based on common columns.

```
SUBTRACT tblname1 FROM tblname2 FORMING [ TEMPORARY  
INMEMORY ] tblname3 [ USING collist ]
```

Options

FORMING tblname3

Specifies the name of the table that is produced when you subtract one table from another.

FROM tblname2

Specifies the name of the table from which you want to subtract another table.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using INMEMORY in place of TEMPORARY creates the temporary table in memory, rather than a temporary file.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

tblname1

Specifies the name of the table you want to subtract from a second table.

USING collist

Specifies the columns and order of columns that you want to include in the new table. At least one column in the USING clause must exist in both tables. If you do not include the clause, R:BASE makes a new table that includes all columns from both tables.

About the SUBTRACT Command

When one table (*tblname1*) is subtracted from another table (*tblname2*), R:BASE looks for columns in both tables that have the same name. The values in these common columns are subtracted and placed in the new table (*tblname3*). *Tblname3* also contains all columns from *tblname2* that are not in common with *tblname1*, unless columns are listed in the USING clause. It is worth noting that when there is more than one column with the same name R:BASE will not allow you to use the SUBTRACT command without using the USING clause to specify which columns to compare.

Use SUBTRACT when you want to create a table that identifies differences between two tables, such as unique rows. When computed columns are common, R:BASE subtracts the current values in the computed columns. If you include a computed column in the USING clause, R:BASE transfers the current value for the computed column to the new table. If you want a computed column in a new table, include

each column used in the computed column's expression with a USING clause. If you do not, R:BASE changes the computed column to a regular column containing data.

In a new table, R:BASE does not update the value in a row for an autonumbered column. An autonumbered column becomes a regular column.

You can increase the speed of SUBTRACT by creating an index for a common column in *tblname1*. SUBTRACT does not use an index in *tblname2*. If both tables have multiple common columns, index the common column in *tblname1* that is most distinctive.

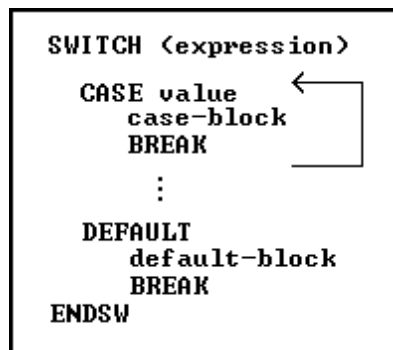
Example

The following command creates a new table named *nobonus* by subtracting the common columns in the *salesbonus* table from the *employee* table.

```
SUBTRACT salesbonus FROM employee FORMING nobonus
```

1.20.23 SWITCH/ENDSW

Use the SWITCH...ENDSW command in a program to define a block of possible actions to take depending on the value of an expression. The SWITCH and CASE statements help control complex conditional and branching operations.



Options

BREAK

Ends SWITCH processing; use this option within each CASE comparison and in the DEFAULT block.

case-block

Contains one or more commands to execute if the CASE value matches the SWITCH expression.

CASE value

Compares the SWITCH value to another value. If the values match, the commands following the CASE are executed; otherwise, the next CASE comparison is checked.

DEFAULT

Provides commands to execute if no CASE comparisons are true.

default-block

Contains one or more commands to execute if no CASE comparisons are true.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

About the SWITCH...ENDSW Command Structure

The SWITCH statement is a control statement that handles multiple selections by passing control to one of the CASE statements within its body whose value matches the initial expression. The switch statement

transfers control to a statement within its body. The syntax diagram shows the entire SWITCH...ENDSW, including the SWITCH value, CASE blocks, and the DEFAULT block.

```
SWITCH ([expression])
    CASE [value]
        [case-block]
        BREAK
DEFAULT
    [default-block]
    BREAK
ENDSW
```

The SWITCH statement allows a developer to control the order in which the code is executed because it is a conditional statement. In addition to handling multiple selections by passing control to one of the CASE statements within its body, the SWITCH provides an efficient mechanism for controlling, tracing, and debugging output at run time using external settings. Practically, you may define a common command file template/Custom Form Action and then use the R:BASE percent variables (%) to pass all required parameters for SWITCH and CASE values. A nested [IF...ENDIF](#) can be rewritten as one SWITCH statement.

The SWITCH Expression

SWITCH defines the expression to be compared. You can have multiple comparisons, so ENDSW defines the end of the comparisons. The SWITCH expression result must be either an INTEGER or a TEXT data type. The SWITCH expression can be a calculation, constant value, or variable. Any length of text can be compared, but only the first 30 characters are checked in each CASE block.

CASE Blocks

The SWITCH statement can include any number of CASE instances, but no two case statements can have the same value. A CASE block consists of three parts: the CASE comparison, the commands following each comparison, and the BREAK statement.

CASE comparisons must be the same data type as the SWITCH expression result - either INTEGER or TEXT. A CASE value cannot be an expression, but must be a constant value or a variable. You can have multiple CASE comparisons to run a single set of commands. For an example of how to use multiple comparisons, see "Examples" below.

The commands following a CASE comparison can include any R:BASE command, including a nested SWITCH...ENDSW structure. You can nest as many SWITCH...ENDSW structures as memory allows.

Use a BREAK statement as the last command in a CASE block to exit from the SWITCH...ENDSW structure. The BREAK command stops R:BASE from checking any additional CASE comparisons. Otherwise, R:BASE will execute every CASE statement even if only one condition is met.

The DEFAULT Block

Use the DEFAULT block to provide a set of commands to be executed if none of the CASE comparisons matches the expression. You can have only one DEFAULT block for each SWITCH...ENDSW structure. The DEFAULT block should be located in the last statement block in a SWITCH...ENDSW structure. If a CASE block follows a DEFAULT block, R:BASE generates a warning.

Example

The following SWITCH...ENDSW structure uses a date entered in a [DIALOG](#) command in the expression. The TDWK function calculates day of the week as text from the date stored in *vday*.

```
DIALOG 'Enter a date:' vday vendkey 1
SWITCH (TDWK(.vday))
    CASE 'Saturday'
    CASE 'Sunday'
        WRITE 'This is a weekend day.'
    SHOW VARIABLE vday
```

```

        BREAK
    DEFAULT
        WRITE 'This is a weekday.'
        SHOW VARIABLE vday
    BREAK
ENDSW

```

If you entered 12/17/94 when prompted for the date, the first CASE comparison would check whether the day of the week is the word *Saturday*. Because the word is *Saturday*, R:BASE would display the message below. The BREAK command prevents R:BASE from processing the rest of the commands in the SWITCH...ENDSW structure.

```

This is a weekend day.
12/17/94

```

If the date entered is not *Saturday* or *Sunday* - for example, 12/22/94 - the information in the DEFAULT block would display the following.

```

This is a weekday.
12/22/94

```

1.21 T

1.21.1 TALLY

Use the TALLY command to count the occurrence of identical values in a column or expression.

```

TALLY { colname | expression } FROM tblview [ WHERE clause ]

```

Options

colname

Specifies the name of the column from which you want to tally information.

expression

Determines a value using a text or arithmetic formula. The expression must be a value that can be evaluated from looking at data in a single row. Lookup expressions are not supported.

FROM tblview

Specifies the table or view.

WHERE clause

Limits rows of data. See [WHERE](#).

About the TALLY Command

The TALLY command displays each value and its number of occurrences as a list. Identical values can also be counted by choosing **Calculate: Tally** from the Data Browser.

For the TEXT data type, the TALLY command evaluates the first 100 characters. TALLY does not work with the NOTE data type.

To display data without column headings, set [HEADINGS](#) off before using the TALLY command. Setting HEADINGS off is useful if you direct output of the TALLY command to a device (printer or file) other than the screen, for example, to a file that you intend to load into another database. The default setting for HEADINGS is on.

Use expressions and functions to perform calculations whose results appear as a column for the occurrence count result.

Examples

Example 01:

The following command lists occurrences of each customer identification number in the *TransMaster* table. R:BASE sorts the information in ascending order by *Custid*.

```
TALLY CustID FROM TransMaster
```

Example 02:

The following command lists the integer day of the month occurrences for each transaction date in the *InvoiceHeader* table using the IDAY function. R:BASE sorts the information in ascending order by the day of the month.

```
TALLY (IDAY(Transdate)) FROM InvoiceHeader
```

```
(IDAY(Transdate)) Number of Occurrences
```

1	12
2	6
3	4
5	2
6	6
7	2
9	2
10	2
11	1
12	5
13	1
14	1
15	6
16	2
17	2
18	1
19	3
20	2
21	3
22	4
23	8
25	7
26	1
27	6
28	2
29	2
30	4

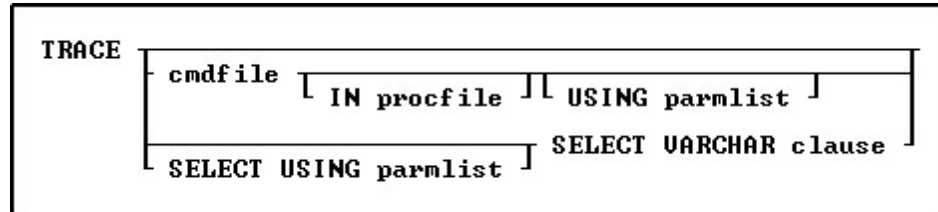
1.21.2 TILE

Use the TILE command to tile the open windows. TILE was used in earlier Windows versions of R:BASE with MDI. The command is no longer supported.

TILE

1.21.3 TRACE

Use the TRACE command to debug command blocks, command files, or command files requiring passed parameters. [Ctrl] + [Shift] + [F12] will trace the next entry/exit procedure.



Options

cmdfile

Specifies the name of the command file to debug.

On a workstation with multiple drives (local or mapped), especially when the files are on the different drive, it is always the best practice to define a drive letter when copying, deleting, renaming, or running files, unless the specified files are located in the working directory. You will not need to specify the drive letter if all of the files are located in the default directory when using file-based commands.

IN procfile

Specifies the name of a procedure file. A procedure file is a compiled binary file that contains stored menu, screen, and command blocks. Include a drive and path name if the procedure file is not on the current drive and directory.

SELECT VARCHAR clause

Specifies a column defined with the VARCHAR data type from a table, then you can run the contents. The [SELECT](#) clause must limit the data to only one row; otherwise, an error is returned.

USING parmlist

Lists the values the command file will use when it executes. The parameter list can contain up to nine values. The first value in the list is referenced in the executed file as %1, the second as %2, and so on through %9. They are treated just like other variables. To reference the contents of these variables, preface the variable name with a dot (.); for example, set v1 = .%1.

About the TRACE Command

The TRACE command starts the R:BASE interactive debugger. It allows you to go through the command file line-by-line to facilitate debugging the code. The TRACE command must be on a line by itself and not combined with other commands. For TRACE to work correctly with a file encoded by CodeLock, the ASCII file must be present in the current drive and directory, and have the file extension of .APP. If TRACE cannot find the ASCII file, it does not go through the procedure file line-by-line when it runs.

Immediately after you find an error in the R:BASE debugger, you can correct bad syntax using the R:BASE Editor. To correct an error:

1. If an error point is found, the debugger displays an error message and prompts to modify the file.
2. If you wish to modify it at once, click the Yes button. The R:BASE Editor will then open, and focus is placed upon the error point.
3. Correct the syntax error, save the file, and exit the R:BASE Editor.
4. To repeat debugging, restart the debugger using the TRACE command.

For more information about the interactive debugger, refer to the TRACE compiled Help in the R:BASE program directory.

Examples

The following command displays the MYCMDFIL file for debugging in the current directory.

```
TRACE mycmdfil
```

The following command displays a command block for debugging named *mycmdfil* in a procedure file named MYPROCFL.APX.

```
TRACE mycmdfil IN myprocfl.apx
```

The following TRACE command displays the Trace Debugger for the command syntax with the *CmdData* VARCHAR data field inside the table *IntrnlCmd*.

```
TRACE SELECT CmdData FROM IntrnlCmd WHERE CmdName = 'NewKey'
```

TRACE can also be initiated inside a command file to trace a block of code using the [SET TRACE ON](#) operating condition.

1.21.4 TURBO

Use the TURBO command to convert your database files to R:BASE 11 database files.

TURBO dbname [**v9**] [IDENTIFIED BY [OWNER]]

Options

dbname

Specifies a database name

v9

Specifies that you are converting a R:BASE Turbo V-8 database to R:BASE 11.

IDENTIFIED BY

Specifies the user identifier. If left blank, R:BASE prompts you for the user identifier. R:BASE does not display it as you enter the text.

OWNER

Optional; specifies the database owner name. If omitted and an OWNER name exists, you will be prompted.

Notes:

- If you are converting a RB1-RB4 database file structure to RX1-RX4 for the first time, you do not need to specify the "V9" parameter.
- The TURBO command will convert database files as far back as version 6.5.

For more on database conversions, please refer to the Database Conversion Guide PDF document located within the R:BASE program directory).

1.21.5 TYPE

Use the TYPE command to send an ANSI or Unicode file to another file, screen, or printer.

TYPE filespec

Option

filespec

Specifies a file name with a drive and path specification in the form D:\PATHNAME\FILENAME.EXT.

About the TYPE Command

The R:BASE TYPE command is similar to the operating system command-line interface TYPE command. The R:BASE TYPE command is different in that it displays a screen of data and prompts you to press any key to continue. In R:BASE, you can use TYPE to display a file and review its contents. If TYPE is used in an R:BASE command file, it can display any ANSI or Unicode file.

The [SET WIDTH](#) and [SET LINES](#) conditions effect how TYPE displays data. For example, if you set width to 40 characters and lines to 15, TYPE displays 15 lines of data before the "Press any key to continue" message displays, and wraps each line of data after 40 characters.

To send a file to a printer, enter [OUTPUT PRINTER](#) before the TYPE command.

Examples

The following command displays the file named TestFile.dat stored on the current drive and directory.

```
TYPE TestFile.dat
```

The following command displays the file named CustomerData.dat stored in the CustDir directory on drive D:\.

```
TYPE D:\CustDir\CustomerData.dat
```

The following command prints the file named NewCustFile.txt stored on drive C:\, then returns output to the screen.

```
OUTPUT PRINTER
TYPE C:\NewCustFile.txt
OUTPUT SCREEN
```

1.22 U

1.22.1 UNION

Use the UNION command to combine the columns and rows of two tables into a new table.

```
UNION tblname1 WITH tblname2 FORMING [ TEMPORARY
INMEMORY ] tblname3 [ USING collist ]
```

Options

FORMING tblname3

Specifies the name of the new table that is produced.

INMEMORY

Specifies to create a temporary table in memory that disappears when the database is disconnected. Using INMEMORY in place of TEMPORARY creates the temporary table in memory, rather than a temporary file.

TEMPORARY

Creates a temporary table that disappears when the database is disconnected.

tblname1

Specifies the name of one of two tables you will combine.

USING collist

Specifies the columns and order of columns that you want to include in the new table. At least one column in the USING clause must exist in both tables. If you do not include the clause, R:BASE makes a new table that includes all columns from both tables.

WITH tblname2

Specifies the name of the second of two tables you will combine.

About the UNION Command

When two tables are added together, R:BASE looks for columns in both tables that have the same name. The values in these common columns are added together and placed in the new table.

If the common columns have more than one match, R:BASE adds a new row for each possible combination. For example, if a table with a column named *empid* has two rows with an identification number of 100, and a second table has three rows with the same identification number, the new table will have six rows—one for each combination.

You can increase the speed of UNION by creating an index for the common column in both tables. If the tables have multiple common columns, index the common column that has the greatest number of unique values.

Combining Various Column Types

When computed columns are common columns, R:BASE compares the current values and changes the computed columns to regular columns in the new table. If you want a computed column in the new table, you should include each column used in the computed column's expression before the computed column.

If you use the UNION command to combine one or more columns that have NOTE data types, R:BASE truncates the columns that have the NOTE data types until the total row length is within the 32,768-character limit.

In a new table, R:BASE does not update the value in a row for an autonumbered column. An autonumber column becomes a regular column.

Creating New Columns

You can also use UNION to add new columns to a table. Define the new columns in a temporary table with the [CREATE TABLE](#) command. Then, combine the existing table with the temporary table to form a new table containing all the columns. When the union is complete, use the [DROP](#) command to delete the temporary table. The temporary table must have at least one column in common with the existing table.

Examples

The following command creates the *saleshist* table by combining all rows for the *empid*, *empfname*, *emplname*, and *netamount* columns. The *saleshist* table contains the columns listed in the USING clause in the order given.

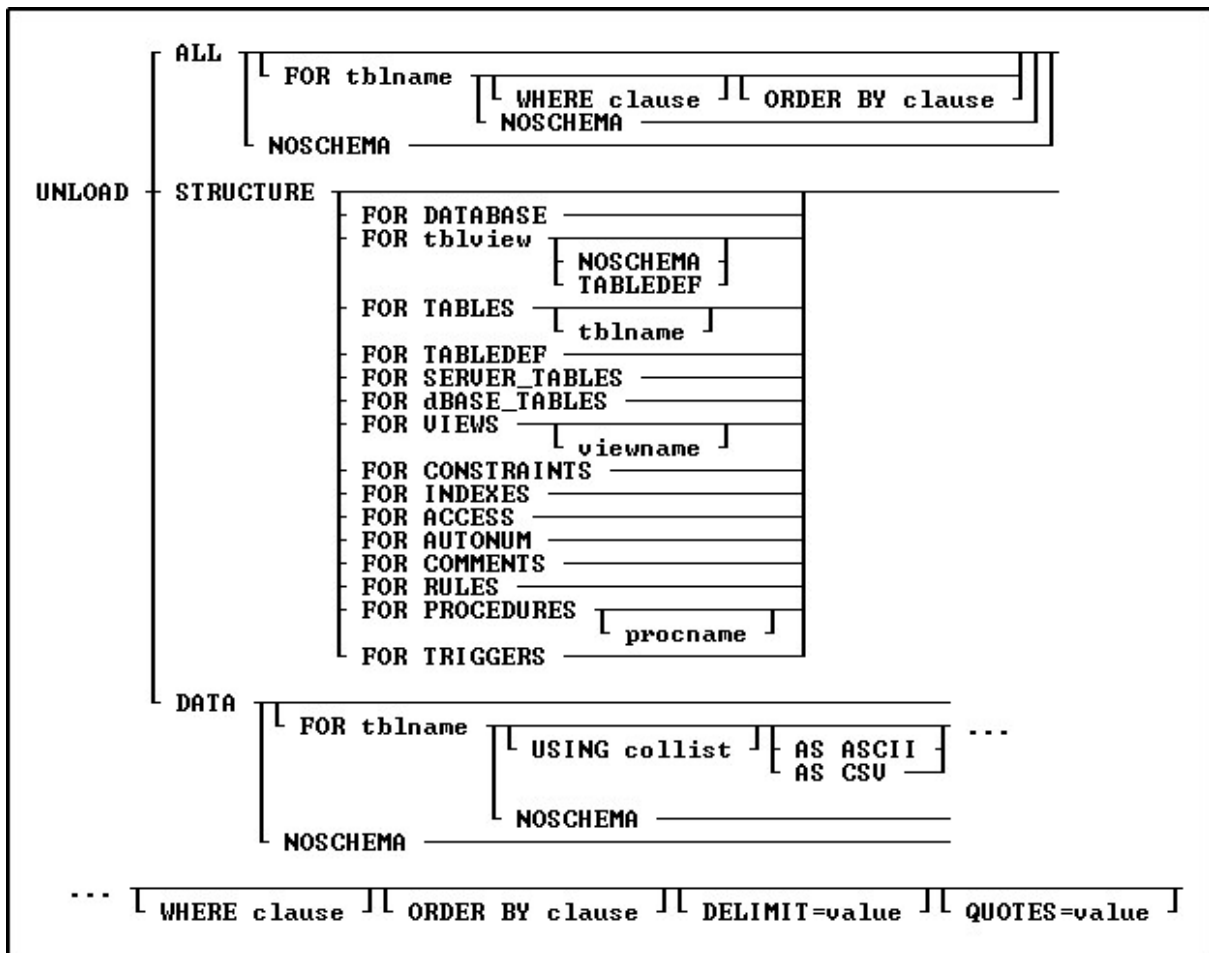
```
UNION employee WITH transmaster FORMING saleshist USING +
empid, empfname, emplname, netamount
```

The following command creates the *reporthistory* table by combining all columns and rows from the *employee* and *salesbonus* tables. If a USING clause is not specified, *reporthistory* includes all the columns from the *employee* and *salesbonus* tables, listing the columns from the *employee* table first.

```
UNION employee WITH salesbonus FORMING reporthistory
```

1.22.2 UNLOAD

Use the UNLOAD command to copy the data, structure, or data and structure of a database or table to a specified output device.



Options

ALL

Unloads both the data and the structure.

AS ASCII

Unloads data in ASCII-delimited format. Use only with the UNLOAD DATA command.

AS CSV

Unloads data in a minimally quoted comma separated format. Each field will be separated by the current DELIMIT character (usually the comma). Fields that contain the current DELIMIT character will be enclosed in the current QUOTES character.

DATA

Unloads the table data. For a database unload, the data includes table data and system table data.

DELIMIT=value

Specifies a custom delimiter value for the ASCII or CSV unload formats. The value can be a character or the CHAR function.

FOR ACCESS

Unloads all current access rights in the database.

FOR AUTONUM

Unloads the [AUTONUM](#) column formula for all autonumbered database columns in the database.

FOR COMMENTS

Unloads all comments assigned to tables, views, and columns in the database.

FOR CONSTRAINTS

Unloads all primary key, foreign key, and unique key relational constraints in the database, based on their order of creation.

FOR DATABASE

Unloads the character settings, the CREATE SCHEMA command, database [comment](#), and [static variables](#).

FOR DBASE_TABLES

Unloads the structure for all attached dBASE tables. To unload the structure for a single dBASE table, use the UNLOAD STRUCTURE FOR tblname command.

FOR INDEXES

Unloads all indexes in the database.

FOR PROCEDURES

Unloads all defined stored procedures into individual files. A .RMD and .PRO file is created for each stored procedure.

FOR SERVER_TABLES

Unloads the structure for all ODBC attached Server tables. To unload the structure for a single Server table, use the UNLOAD STRUCTURE FOR tblname command.

FOR TABLES

Unloads the complete table structure for all tables in the database. The output includes the table definitions, [last modification date/time stamp](#), relational constraints; based on their order of creation, access rights, autonumbered columns, comments, rules, and triggers. Wildcards can be used to unload the structure for multiple tables/views starting with the same letters (i.e UNLOAD STRUCTURE FOR TABLES CUST%).

FOR TABLEDEF

Unloads the table definition structure and last modification date/time stamp for all tables in the database. The output contains only the SQL command necessary to create all the tables, including SCONNECT and SATTACH commands for Server tables.

FOR tblview

Specifies a single table/view to unload the SQL command necessary to create a specific table/view. The output includes the table definition, last modification date/time stamp, relational constraints; based on their order of creation, access rights, autonumbered columns, comments, rules, and triggers.

FOR TRIGGERS

Unloads all triggers in the database.

FOR RULES

Unloads all rules in the database.

FOR VIEWS

Unloads the SQL command necessary to create all views. The output includes the view definition and the [last modification date/time stamp](#),

NOSCHEMA

Omits the character settings and [CREATE SCHEMA](#) syntax from the beginning of the output.

ORDER BY clause

Sorts rows of data. For more information, see [ORDER BY](#).

procname

Specifies the procedure name.

QUOTES=value

Specifies a custom quote character value for the ASCII or CSV unload formats. The value can be a character or the CHAR function.

STRUCTURE

Unloads the database structure, in the following order: schema, database [comment](#), [static variables](#), tables, views, constraints/indexes, users, privileges, autonumber definitions, comments, rules, stored procedure data, triggers, and database events.

STRUCTURE FOR tblview

Unloads the SQL commands necessary to create a specific table/view. The table output includes the table definition, [last modification date/time stamp](#), relational constraints; based on their order of creation, access rights, autonumbered columns, comments, rules, and triggers. The view output includes the structure and the last modification date/time stamp.

TABLEDEF

Unloads the table/view definition structure. The output contains only the SQL command necessary to create the table/view and the last modification date/time stamp.

tblname

Specifies the table name to unload the data, structure, or both.

USING collist

Specifies the column(s) to use with the command. When using ASCII or CSV unloads, text may also be inserted.

```
UNLOAD DATA FOR Contact USING 'Contact Name:',ContFName,ContLName AS CSV
```

viewname

Specifies the view name to unload the structure and the [last modification date/time stamp](#). Wildcard are supported to specify multiple views starting with the same letters (i.e UNLOAD STRUCTURE FOR VIEWS Sale%).

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

About the UNLOAD Command

Use UNLOAD to transfer tables or views from one database to another, or to backup a database. Forms, reports, labels, and stored procedures may also be unloaded.

The UNLOAD ALL and UNLOAD STRUCTURE commands require the database owner's user identifier if the database has had access rights granted with the [GRANT](#) command. R:BASE places the owner's user identifier and all the granted access rights in the file created by UNLOAD to ensure that the restored database file continues to be protected. Be sure to protect this file from unauthorized users.

The UNLOAD command creates a file with a .LOB extension for binary large objects, and the originating file that you specify for the data and/or structure. Your originating file can NOT have a .LOB file extension, otherwise, R:BASE will not be able to continue with the UNLOAD process.

The UNLOAD command unloads defined database events ([ON CONNECT/PACK](#), [ON ERROR](#), [ON DISCONNECT/EXIT](#)) if an UNLOAD ALL or UNLOAD STRUCTURE is performed.

Transferring Tables and Views

UNLOAD does not change the data or structure of the original database, and it does not unload computed column values unless you use the AS ASCII or AS CSV option.

You can also unload and transfer the data and structure of a view. If a table or view needed to define a view cannot be found, after the error message about the missing item, you will get another message to identify the view that R:BASE was trying to create. To input the data from an unloaded view into a table, create a table to match the view's structure and use the [LOAD tblname FROM filespec](#) command, since views do not contain data. The UNLOAD command is useful when you want to create a file to transfer data to another database as a table. Within the unloaded table and view structure output is the [RESET](#) command, which preserves the last modification date/time stamp.

To be able to reliably load data back into R:BASE from an unloaded file:

- [SET NULL](#) to -0- (the R:BASE default) before unloading data.
- Do not set a special character to the same setting as another special character.

- Precede UNLOAD with an [OUTPUT](#) command to direct the data to be unloaded to a file. You can edit the unloaded file with any ASCII text editor.
- Ensure the [IDQUOTES](#) setting for the unloaded output is identical to the destination database.
- Use the DELIMIT and QUOTES options to specify custom values.

To transfer unloaded information back into R:BASE, use the [RUN](#), [RESTORE](#), or [GATEWAY](#) commands, if transferring only data. For example, you can use the UNLOAD DATA command to unload data, then the RUN command to transfer the data to a different database.

If the UNLOAD AS CSV syntax has been used you can use the [LOAD AS CSV](#) command to restore the data.

Backing up a Database

R:BASE unloads data in ASCII delimited format: values are separated by the current delimiter and all text strings are enclosed in quotation marks. UNLOAD creates a file containing commands that set special characters, such as commas and quotation marks. The setting of the [SET WIDTH](#) condition affects the width of data lines in the unloaded file.

If the database has columns defined as binary or text large objects, then UNLOAD creates two files, one file containing the R:BASE commands, and a second file (with a .LOB extension) containing the large object data. Both files are needed to transfer the information back into R:BASE. Your originating file can NOT have a .LOB file extension, otherwise, R:BASE will not be able to continue with the UNLOAD process.

Note: The unloaded file cannot span multiple floppy disks.

The UNLOAD STRUCTURE or UNLOAD ALL commands write all the commands necessary to define the database or table, starting with [CREATE SCHEMA](#) AUTHORIZATION *dbname* near the beginning of the file. Before you input or restore the data or structure into a different database, you can use a text editor to change the database name in the unloaded file. If you use UNLOAD STRUCTURE, you can copy the table structure after you change the database name in the unloaded file.

Unloading Temporary Tables

If there are temporary tables/views in a database when an UNLOAD ALL or UNLOAD STRUCTURE is performed, the temporary table/view definitions will be unloaded.

Use the UNLOAD *tblname* command to backup individual temporary tables created when [STATICDB](#) is set on-which activates a read-only schema mode. When UNLOAD is used to backup temporary tables, it generates a SET STATICDB OFF command to be executed prior to the CREATE SCHEMA command.

Note:

- UNLOAD ALL and UNLOAD STRUCTURE will include the definitions for temporary tables/views.
- Wildcards can be used to unload the structure for multiple tables/views starting with the same letters (i.e UNLOAD STRUCTURE FOR TABLES CUST%).

Unloading Computed Columns

When the original computed values must be preserved with an UNLOAD and LOAD, the [NOCALC](#) setting must be set ON. The UNLOAD output generated will also have a NOCALC after the LOAD command so that the original values are loaded.

Examples

Example 01:

The following command lines unload only the data from the *product* table to a file named MYFILE.DBS. The data is in ASCII delimited format. The [OUTPUT SCREEN](#) command redirects the output back to the screen and closes the file.

```
OUTPUT myfile.dbs
UNLOAD DATA FOR product AS ASCII
OUTPUT SCREEN
```


Example 02:

In the example below the a file will be created that contains Comma Separated Values with no headings and no page breaks.

```
SET HEADINGS OFF
SET LINES 0
SET WIDTH 200
OUTPUT myfile.csv
UNLOAD DATA FOR Employees AS CSV
OUTPUT SCREEN
```

The commands above might create the file below. Notice that Jane Dough has Quotes surrounding her address. This is because the text contains an embedded comma.

```
Robert,Smith,123 Main St,Denver,CO,Support
Jane,Dough,'98 Folk St, Apt 1',Pittsburgh,PA,Sales
Matt,Follows,14 Arrowhead Ln,Portsmouth,RI,Services
```

Example 03:

The following command unloads the complete structure for the *Contact* table within the *RRBYW20* sample database.

```
R>UNLOAD STRUCTURE FOR Contact

SET QUOTES=NULL
SET QUOTES='
SET DELIMIT=NULL
SET DELIMIT=', '
SET LINEEND=NULL
SET LINEEND='^'
SET SEMI=NULL
SET SEMI=';'
SET PLUS=NULL
SET PLUS='+'
SET SINGLE=NULL
SET SINGLE='_ '
SET MANY=NULL
SET MANY='% '
SET IDQUOTES=NULL
SET IDQUOTES='`'
SET CURRENCY '$' PREF 2 B
DISCONNECT
SET STATICDB OFF
SET ROWLOCKS ON
SET FASTLOCK OFF
CREATE SCHEMA AUTHOR RRBYW20 NONE
CREATE TABLE `Contact` +
(`CustID` INTEGER, +
`ContID` INTEGER NOT NULL +
('Value for column ContID cannot be null') , +
`ContFName` TEXT (10), +
`ContLName` TEXT (16), +
`ContPhone` TEXT (14), +
`ContFax` TEXT (14), +
`ContCell` TEXT (14), +
`ContPager` TEXT (14), +
`ContEMail` TEXT (60), +
`ContInfo` NOTE, +
```

```

`ContPhoto` LONG VARBIT, +
`LastContactDate` DATE)
RESET Contact LAST_MOD TO '10/06/2018 21:24:51'
ALTER TABLE `Contact` ADD PRIMARY KEY (`ContID` ) +
('Values for rows in Contact must be unique',+
'Cannot delete - values exist in another table',+
'Cannot update - values exist in another table')
ALTER TABLE `Contact` ADD FOREIGN KEY +
( `CustID` )+
REFERENCES `Customer`
AUTONUM `ContID` IN `Contact` USING 1047. 1. NONUM
COMMENT ON `CustID` IN `Contact` IS 'Customer Identification Number'
COMMENT ON `ContID` IN `Contact` IS 'Contact Automated ID'
COMMENT ON `ContFName` IN `Contact` IS 'Customer Contact First Name'
COMMENT ON `ContLName` IN `Contact` IS 'Customer Contact Last Name'
COMMENT ON `ContPhone` IN `Contact` IS 'Customer Contact Phone Number'
COMMENT ON `ContFax` IN `Contact` IS 'Customer Contact Fax Number'
COMMENT ON `ContCell` IN `Contact` IS 'Customer Contact Cellular Number'
COMMENT ON `ContPager` IN `Contact` IS 'Customer Contact Pager Number'
COMMENT ON `ContEMail` IN `Contact` IS 'Customer Contact E-Mail Address'
COMMENT ON `ContInfo` IN `Contact` IS 'Customer Contact Notes'
COMMENT ON `ContPhoto` IN `Contact` IS 'Customer Contact Photo'
COMMENT ON `LastContactDate` IN `Contact` IS 'Last Contact Date'
COMMENT ON TABLE `Contact` IS 'Customer Contact Information'

```

Example 04:

The following command unloads the structure for the *Contact* table within the *RRBYW20* sample database, without any database schema at the beginning.

```

R>UNLOAD STRUCTURE FOR contact NOSHEMA

CREATE TABLE `Contact` +
(`CustID` INTEGER, +
`ContID` INTEGER NOT NULL +
('Value for column ContID cannot be null') , +
`ContFName` TEXT (10), +
`ContLName` TEXT (16), +
`ContPhone` TEXT (14), +
`ContFax` TEXT (14), +
`ContCell` TEXT (14), +
`ContPager` TEXT (14), +
`ContEMail` TEXT (60), +
`ContInfo` NOTE, +
`ContPhoto` LONG VARBIT, +
`LastContactDate` DATE)
RESET Contact LAST_MOD TO '10/06/2018 21:24:51'
ALTER TABLE `Contact` ADD PRIMARY KEY (`ContID` ) +
('Values for rows in Contact must be unique',+
'Cannot delete - values exist in another table',+
'Cannot update - values exist in another table')
ALTER TABLE `Contact` ADD FOREIGN KEY +
( `CustID` ) +
REFERENCES `Customer`
AUTONUM `ContID` IN `Contact` USING 1047. 1. NONUM
COMMENT ON `CustID` IN `Contact` IS 'Customer Identification Number'
COMMENT ON `ContID` IN `Contact` IS 'Contact Automated ID'
COMMENT ON `ContFName` IN `Contact` IS 'Customer Contact First Name'

```

```
COMMENT ON `ContLName` IN `Contact` IS 'Customer Contact Last Name'
COMMENT ON `ContPhone` IN `Contact` IS 'Customer Contact Phone Number'
COMMENT ON `ContFax` IN `Contact` IS 'Customer Contact Fax Number'
COMMENT ON `ContCell` IN `Contact` IS 'Customer Contact Cellular Number'
COMMENT ON `ContPager` IN `Contact` IS 'Customer Contact Pager Number'
COMMENT ON `ContEMail` IN `Contact` IS 'Customer Contact E-Mail Address'
COMMENT ON `ContInfo` IN `Contact` IS 'Customer Contact Notes'
COMMENT ON `ContPhoto` IN `Contact` IS 'Customer Contact Photo'
COMMENT ON `LastContactDate` IN `Contact` IS 'Last Contact Date'
COMMENT ON TABLE `Contact` IS 'Customer Contact Information'
```

Example 05:

The following command unloads the indexes for the *ConComp* sample database.

```
R>UNLOAD STRUCTURE FOR INDEXES
```

```
CREATE INDEX CustState ON `Customer` +
(`CustState` ASC SIZE 2 )
```

Example 06:

The following command unloads the rules for the *ConComp* sample database.

```
R>UNLOAD STRUCTURE FOR RULES
```

```
RULES 'Value for onhand cannot be less than minimum.' +
      FOR `ProdLocation` SUCCEEDS +
      WHERE ProdLocation.Onhand >= 1
```

Example 07:

The following unloads the first name, last name, and the preceded text "Contact Name" with the ASCII format

```
UNLOAD DATA FOR Contact USING 'Contact Name:',ContFName,ContLName AS ASCII
```

Example 08:

The following unloads data from the InvoiceHeader table with the CSV format and uses the pipe character as a custom delimiter

```
UNLOAD DATA FOR InvoiceHeader AS CSV DELIMIT=|
```

Example 09:

The following performs a database unload into three separate parts, to evaluate the rebuild process for flaws in the database stability/integrity. Each DB_Part file can be run or traced individually to monitor the database creation process.

```
-- Unload the database and table structure as Part 1
```

```
OUTPUT DB_Part1.rmd
```

```
UNLOAD STRUCTURE FOR DATABASE
```

```
UNLOAD STRUCTURE FOR TABLEDEF
```

```
OUTPUT SCREEN
```

```
-- Unload the data from the tables as Part 2
```

```
OUTPUT DB_Part2.rmd
```

```
UNLOAD DATA NOSHEMA
```

```
OUTPUT SCREEN
```

```
-- Unload the remaining structure as Part 3
```

```
OUTPUT DB_Part3.rmd
```

```
UNLOAD STRUCTURE FOR VIEWS
```

```
UNLOAD STRUCTURE FOR CONSTRAINTS
```

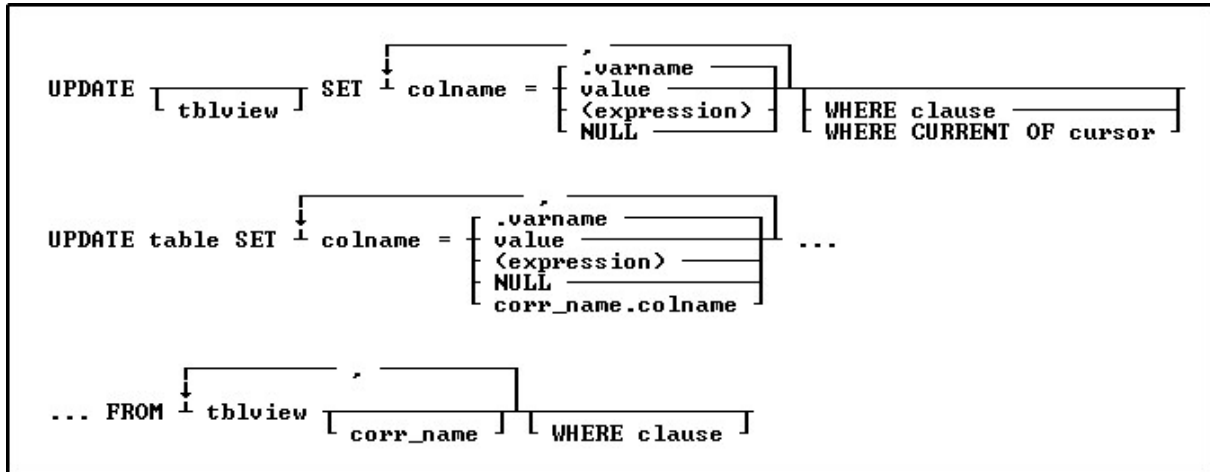
```
UNLOAD STRUCTURE FOR INDEXES
```

```
UNLOAD STRUCTURE FOR ACCESS
```

UNLOAD STRUCTURE FOR AUTONUM
 UNLOAD STRUCTURE FOR COMMENTS
 UNLOAD STRUCTURE FOR RULES
 UNLOAD STRUCTURE FOR PROCEDURES
 UNLOAD STRUCTURE FOR TRIGGERS
 OUTPUT SCREEN

1.22.3 UPDATE

Use the UPDATE command to change the data in one or more columns in a table or a view.



Options

' Indicates that this part of the command is repeatable.

(expression)

Determines a value using a text or arithmetic formula. The expression can include other columns from the table, constant values, functions, or system variables such as *#date*, *#time*, and *#pi*.

corr_name.

Correlation name. A nickname or alias for a table or view name. Use *corr_name* to refer to the same table twice within the command, or to explicitly specify a column in more than one table. The correlation name must be at least two characters.

FROM tblview

Specifies a list of tables or views from which data can be retrieved and updated.

NULL

Sets the values in the column equal to null.

SET colname

Specifies the column to update.

table

Specifies a table.

tblview

Specifies a table or view. If no table or view name is included, columns will be updated in all tables containing the specified columns, according to the conditions of the WHERE clause.

value

Specifies a value to enter in the specified column.

.varname

Specifies a global variable that provides a value for a column.

WHERE clause

Limits rows of data. For more information, see [WHERE](#).

WHERE CURRENT OF cursor

Specifies a cursor that refers to a specific row to be affected by the UPDATE command. With this option, you must specify *tblview*.

About the UPDATE Command

The UPDATE command is useful for adjusting values in columns that require uniform changes.

The UPDATE command only modifies data in columns in one table or view. You can also update a table by referencing values from another table. You can modify a column's value by doing the following:

- Entering a new value for the column as a constant or variable
- Entering an expression that calculates a value for the column
- Entering a null value

Notes:

- Only users that have been granted rights to update the table(s) or column(s) can run the UPDATE command.
- R:BASE complies with defined [rules](#), even for columns not affected by the update. If an update breaks a rule, the update is not processed.
- You cannot use UPDATE with computed or autonumbered columns. To change a computed column value, change the values in the columns to which the computed column refers.
- The UPDATE command will not update data in a multi-table view (a view based on multiple tables), as the data is not editable.
- A view with a [GROUP BY](#) parameter is also not editable.

Updating Column Values

You can update a column with a specific value. The value you use must meet the requirements of the column's data type, for example, a numeric column cannot be loaded with a text value.

Use the current delimiter character (the default is a comma) to separate each column and its new value from the next column and value.

Use the following guidelines when modifying data with UPDATE:

- Do not embed commas within entries for CURRENCY, DATE, DATETIME, DOUBLE, INTEGER, NUMERIC, or REAL data types. R:BASE automatically inserts commas and the current currency symbol.
- When values for CURRENCY, DOUBLE, NUMERIC, or REAL or data types are decimal fractions, you must enter the decimal point. When values are whole numbers, R:BASE adds a decimal point for you at the end of the number. R:BASE adds zeros for subunits in whole currency values. For example, using the default currency format, R:BASE loads an entry of 1000 as \$1,000.00.
- When values for NOTE or TEXT data types contain commas, you can either enclose the entries within quotes, or use [SET DELIMIT](#) to change the default delimiter (comma) to another character.
- When values for NOTE or TEXT data types contain single quotes ('), and you are using the default QUOTES character ('), use two single quotes (") in the text string. For example, 'Walter Finnegan's order.'
- When values for NOTE or TEXT data types exceed the maximum length of a column, R:BASE truncates the value and adds it to the table. A message is displayed that tells you which row has been truncated.

Using an Expression or Variable

Enclose expressions in parentheses. If you use global variables in an expression, dot the variable (*.varname*). If expressions contain values that have a TEXT data type, enclose the values within quotes. The default QUOTES character is the single quote (').

If you attempt to use a null value in an expression or computed column, the result of the expression is null. However, if you set [ZERO](#) to on, R:BASE treats null values as zeros and processes expressions as if the null value were zero.

Using the WHERE Clause

If an UPDATE command includes a table or view name, you do not need to specify a [WHERE](#) or WHERE CURRENT OF clause. All rows will be updated. If you use a [WHERE CURRENT OF](#) clause, you must include a table or view name in the command.

If you omit a table or view name, you must use a WHERE clause with the UPDATE command so that you do not change values in more rows than you intended to change. The WHERE clause pinpoints the rows you want to change. If any columns exist in more than one table, all occurrences are changed if the column value meets the WHERE clause conditions. Test the WHERE clause by using the [SELECT](#) command before using the clause with UPDATE command. By using a WHERE clause with a SELECT command, you can view the rows you want to change before changing them.

R:BASE takes significantly less time to process a WHERE clause if one of the columns specified in the clause is an indexed column.

Using UPDATE with Transaction Processing

If more than one person at a time executes an UPDATE command and transaction processing is on, R:BASE might not execute the command concurrently. If you hold an UPDATE lock, you can read, modify, or delete any row in a table. R:BASE blocks any additional requests for UPDATE until other SELECT or UPDATE locks are cleared.

Examples

The following command changes values in the *company* and *custphone* columns of the *customer* table for the row where *custid* equals 100.

```
UPDATE customer SET company = 'Quality Computers', +
custphone = '617-341-3762' WHERE custid = 100
```

The following command changes the *invoicetotal* column in the *transmaster* table to the value of the expression (*invoicetotal* * .9) for rows where *transid* is greater than 5000.

```
UPDATE transmaster SET invoicetotal = ( invoicetotal * .9) +
WHERE transid > 5000
```

The following command changes the *listprice* column to the value of the expression (*1.1* * *listprice*) for every row in the *prodlocation* table containing an entry in the *listprice* column.

```
UPDATE prodlocation SET listprice = (1.1 * listprice) +
WHERE listprice IS NOT NULL
```

The following command adds to the set of conditions in the above command. The command below extracts all of the selling prices from the *transdetail* table and requires that *listprice* be changed only if it matches a current selling price in the table.

```
UPDATE product SET listprice = (1.1 * listprice) +
WHERE listprice IS NOT NULL AND model = 'CX3000' +
AND listprice IN (SELECT price FROM transdetail +
WHERE model = 'CX3000')
```

The following command changes the *onhand* column in the *prodlocation* table (specified by cursor *curs1*) to the value of the expression (*onhand* - 100). The changes are made only in the row currently referenced by the cursor.

```
UPDATE prodlocation SET onhand = (onhand - 100) +
```

```
WHERE CURRENT OF curs1
```

The following example shows interactive data updating in an application file. The value of *var1* is used in the expression that is assigned to the *onhand* column of the *prodlocation* table. The UPDATE command changes values in *onhand* to the value of the expression (*onhand* - *.var1*) for all rows containing model numbers that begin with the letter C. The wildcard character % indicates one or more additional characters.

```
SET VARIABLE var1 TEXT
DIALOG 'Enter quantity by which to reduce inventory: ' var1 vend 1
SET VARIABLE var1 INTEGER
UPDATE prodlocation SET onhand = (onhand - .var1) +
WHERE model LIKE 'C%'
```

The following command changes the last names of two employees. This command omits the table name, thereby causing a global change to all tables that meet the WHERE clause criteria.

```
UPDATE SET emplname TO 'Smith-Simpson' WHERE +
(empfname = 'Mary' AND emplname = 'Simpson') OR +
(empfname = 'John' AND emplname = 'Smith')
```

The following example corrects a problem that can occur with an incorrect date sequence setting. For example, assume that you had the date sequence set to a four-digit year when you entered transactions, and you entered dates with a two-digit year (3/1/93). The dates would be stored as 3/1/0093. And, if you wanted the date to be in the 20th century, you could use the UPDATE command to modify the existing dates to 20th century dates by adding 1900 years to each date, with the ADDYR function.

The [SET DATE](#) command makes sure that you are using a four-digit year. The UPDATE command changes all *transdate* values to 20th century dates, where the current value of the column is less than 1/1/1900. The last SET DATE command returns to a two-digit date sequence and format.

```
SET DATE MM/DD/YYYY
UPDATE transmaster SET transdate = (ADDYR(transdate,1900)) +
WHERE transdate < 1/1/1900
SET DATE MM/DD/YY
```

Assume that you wanted to update the *inventory* table with the sum of the units sold from the *orders* table. Because there are many rows in the *orders* table for each part number, you cannot do this directly with the UPDATE command. The [CREATE VIEW](#) command creates a view containing the sum of the units sold from the *orders* table. The UPDATE command updates the *inventory* table by extracting the *totalsold* value from the view named *orders_view* for each part number.

```
CREATE VIEW orders_view (partid,totalsold) AS SELECT +
partid, sum(sold) FROM orders GROUP BY partid

UPDATE inventory SET onhand = (T1.onhand - T2.totalsold) +
FROM inventory T1, orders_view T2 +
WHERE T1.partid = T2.partid
```

1.22.4 UPGRADE

Creates required system tables for the latest R:BASE release. The R:BASE session must be in single-user mode.

UPGRADE

About the UPGRADE command

Use the UPGRADE command to create the required system tables, SYS_FORMS3, SYS_LABELS, SYS_LAYOUTS3, SYS_MERGE and SYS_REPORTS3, for the latest R:BASE release.

Example:

```
DISCONNECT
SET MULTI OFF
CONNECT dbname
UPGRADE
LIST SYS%
DISCONNECT
SET MULTI ON
CONNECT dbname
```

Databases that have already been converted to the current format with above system tables will not need to be converted. However, if any of the above system tables are missing, use the UPGRADE command.

1.23 W

1.23.1 WHENEVER

Use the WHENEVER command in a program to check error conditions and run a set of commands designed to handle the error.

WHENEVER { SQLERROR NOT FOUND } [CONTINUE GOTO lblname]

Options

CONTINUE

Turns off error checking by the WHENEVER command.

GOTO lblname

Passes control to the command following the indicated [LABEL](#) command.

NOT FOUND

Indicates that no rows were found by the command (*sqlcode* equals 100).

SQLERROR

Indicates that a processing error of any type other than *data-not-found* was detected (*sqlcode* is less than zero).

About the WHENEVER Command

Use WHENEVER to trap errors when R:BASE Structured Query Language (SQL) commands are run. WHENEVER traps all SQL commands. For more information about structured query language, see Structured Query Language.

R:BASE places the error value of the specific error that occurs in the R:BASE system variable *sqlcode*. You can check the value of *sqlcode* to determine what action to take. The NOT FOUND errors always have the value 100; other errors are the negative of the R:BASE error code number.

Errors in R:BASE commands-not R:BASE SQL commands- do not affect the value of the *sqlcode* error variable. Use an R:BASE error variable to check for R:BASE command errors.

Using the Options

Enter a **WHENEVER SQLERROR** command at the beginning of a command file or command block to check for any SQL processing error other than *data-not-found* errors. **WHENEVER** applies only to that command or block. The system error variable *sqlcode* is set to the negative value of an error condition.

Enter a **WHENEVER NOT FOUND** command at the beginning of a command file or block to check for conditions such as *No rows satisfy the WHERE clause* or *End-of-data encountered* that occur within a program that is running.

The **CONTINUE** option turns off error checking and negates any previously entered **WHENEVER** command. That is, if you have entered a **WHENEVER NOT FOUND** with a **GOTO**, R:BASE runs the **GOTO** whenever data is not found. Then, if you enter a **WHENEVER NOT FOUND** command with a **CONTINUE**, *data-not-found* errors do not run the **GOTO** action. The **GOTO** action passes control to the specified label. You must have a matching **LABEL** command in the same command file or block within the procedure file running the **WHENEVER** command.

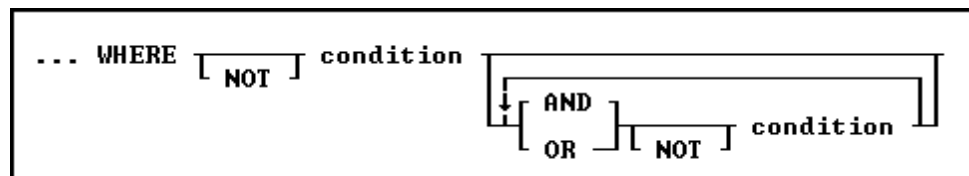
Example

In the following example, the **DECLARE CURSOR** and **OPEN** commands set a route to the *transmaster* table when the value of *transid* matches the value in the *vtran* variable. If the **FETCH** command does not find any rows, then R:BASE passes control to the commands that follow the **LABEL** command. The section after the **LABEL** command should contain commands to check the value of the *sqlcode* system variable and then perform some action according to the specific error condition that occurs.

```
WHENEVER NOT FOUND GOTO errors
.
.
.
DECLARE curl CURSOR FOR SELECT custid, netamount FROM +
    transmaster WHERE transid = .vtran
OPEN curl
FETCH curl INTO vid, vamount
.
.
.
LABEL errors
*(error handling commands)
```

1.23.2 WHERE

Use a **WHERE** clause in commands to qualify or restrict the rows affected by a command.



Options

AND

Indicates the following condition must be met along with the preceding condition.

condition

Identifies requirements to be in the **WHERE** syntax.

NOT

Reverses the meaning of a connecting operator. **AND NOT**, for example, indicates that the first condition must be met and the following condition must not be met.

OR

Indicates the following condition can be met instead of the preceding condition.

About the WHERE Clause

In most commands, a WHERE clause follows the syntax diagram above. The [JOIN](#) command does not use that syntax diagram because it uses only comparison operators in a WHERE clause to compare two columns.

The two main elements in any WHERE clause are conditions and connecting operators.

R:BASE supports "COUNT = LAST" in two different methods. If the entire WHERE clause is "WHERE COUNT = LAST" then R:BASE works like it always has, to quickly fetch the last row of the table. The added functionality is to have other conditions in the WHERE clause, and you want the last row of whatever qualifies.

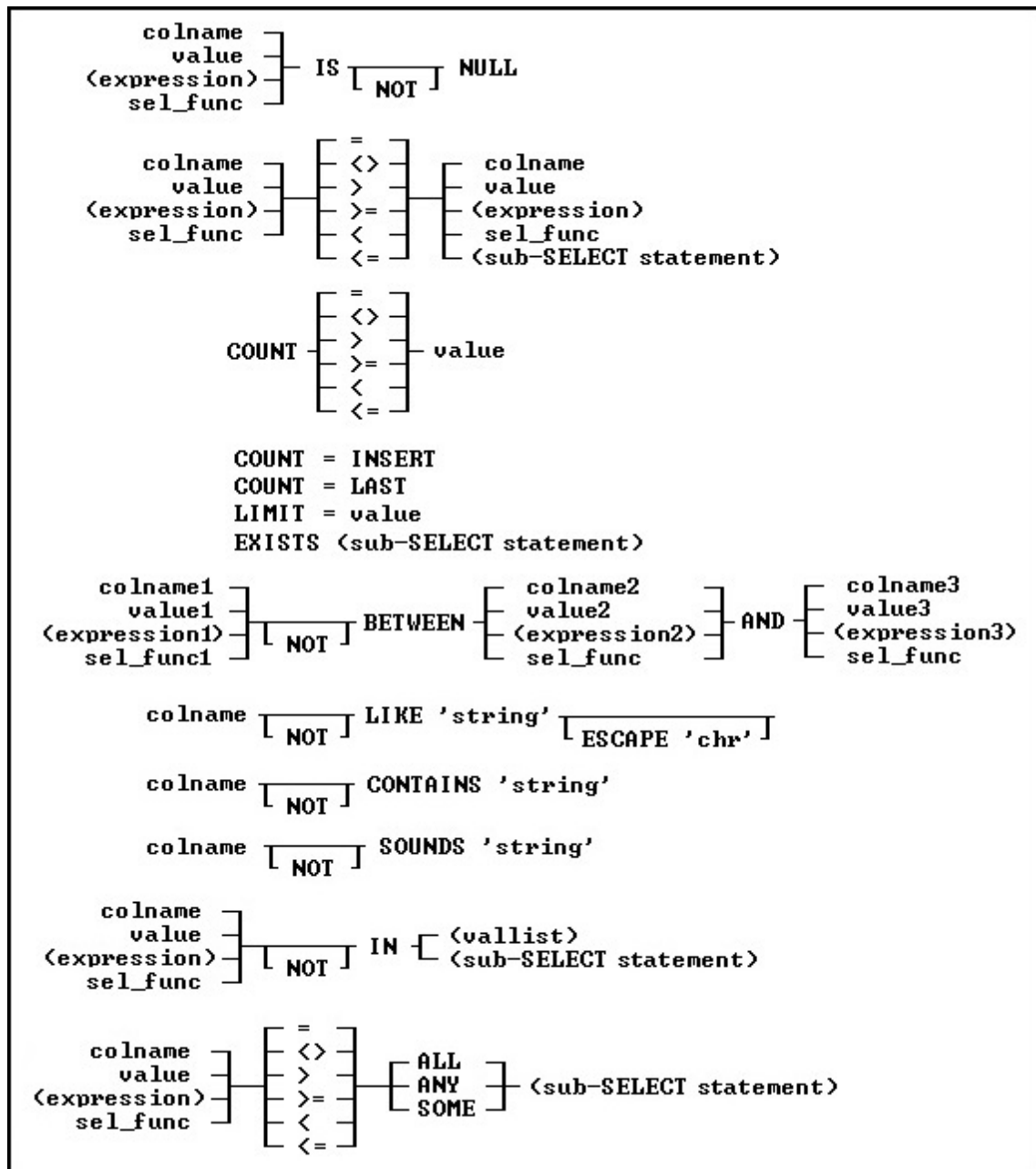
To make it work this way specify the other conditions and then add "AND COUNT = LAST".

Here is an example:

```
SELECT * FROM Customer WHERE CustID > 100 AND COUNT = LAST
```

WHERE Clause Conditions

The following syntax diagram and table show the basic formats for WHERE clause conditions, which can be used alone or together.



Basic WHERE Clause Conditions

Condition Syntax	Description
<i>colname op</i> DEFAULT	True if a column value compares correctly with the DEFAULT value for the column. <i>Op</i> can be =, <>, >=, >, <=, or <.
<i>colname</i> = USER	True if a column value equals the current user identifier.
<i>item1</i> IS NULL	True if <i>item1</i> has a null value. <i>Item1</i> can be a column name, value, or expression. A null value cannot be used in a comparison with an operator.
<i>item1 op item2</i>	True if the relationship between two items is true as defined by an operator. <i>Item1</i> can be a column name, value, or expression; <i>item2</i> can be a column name, value, expression, or sub-SELECT statement.

COUNT=INSERT	Refers to the last row inserted in a table by the current user, even if it has been modified by another user. The COUNT=INSERT condition can be used with a single-table view, but not with a multi-table view. If there is not a newly inserted row in the table, then COUNT=INSERT performs the same action as COUNT=LAST, and fetches the current end row of the table.
COUNT=LAST	Refers to the last row in a table. The COUNT=LAST condition can be used with a single-table view, but not with a multi-table view.
COUNT <i>op value</i>	Refers to a number of rows defined by <i>op</i> and <i>value</i> .
LIMIT= <i>value</i>	Specifies a number of rows affected by a command. A LIMIT condition should be the last condition in a WHERE clause.
EXISTS (sub-SELECT statement)	True if sub-SELECT statement returns one or more rows.
<i>item1</i> BETWEEN <i>item2</i> AND <i>item3</i>	True if the value of <i>item1</i> is greater than or equal to the value of <i>item2</i> , and if the value of <i>item1</i> is less than or equal to the value of <i>item3</i> .
<i>colname</i> LIKE ' <i>string</i> '	True if a column value equals the text string. With LIKE, a string can also be a DATE, TIME, or DATETIME value. The text string can contain R:BASE wildcard characters.
<i>colname</i> LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	True if a column value equals a text string. If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>colname</i> CONTAINS ' <i>string</i> '	True if a column value contains the text string. The text string can contain R:BASE wildcard characters.
<i>colname</i> SOUNDS ' <i>string</i> '	True if the soundex value of a column matches the soundex value of the text string.
<i>item1</i> IN (<i>vallist</i>)	True if <i>item1</i> is in the value list.
<i>item1</i> IN (sub-SELECT statement)	True if <i>item1</i> is in the rows selected by a sub-SELECT.
<i>item1 op</i> ALL (sub-SELECT statement)	True if the relationship between <i>item1</i> and every row returned by a sub-SELECT statement matches an operator.
<i>item1 op</i> ANY(sub-SELECT statement)	True if the relationship between <i>item1</i> and at least one value returned by a sub-SELECT statement matches an operator.
<i>item1 op</i> SOME (sub-SELECT statement)	ANY and SOME are equivalent.

Notes:

- Placing NOT before most text operators (such as NULL or BETWEEN) reverses their meaning.
- When a [SELECT](#) statement is part of a WHERE clause, it is called a [sub-SELECT](#) clause. A sub-SELECT clause can contain only one column name (not a column list or *), expression, or function. The [INTO](#) and [ORDER BY](#) clauses in a sub-SELECT are ignored.

You can only use the current wildcard characters to compare a column to a text value when using the LIKE comparison. The default wildcard characters are the percent sign (%), which is used for one or more characters, and the underscore (_), which is used for a single character.

If you compare a column with a value, you can either enter the value or specify a global variable. If you specify a variable, R:BASE compares the column with the current value of the variable.

To significantly reduce processing time for a WHERE clause, use INDEX processing. To use indexes, the following conditions must be met:

- A condition in the WHERE clause compares an indexed column.
- If the WHERE clause contains more than one condition, R:BASE selects the condition that places the greatest restriction on the WHERE clause.
- Conditions are not joined by the OR operator.
- The comparison value is not an expression.

Connecting Operators

When you use more than one condition in a WHERE clause, the conditions are connected using the connecting operators AND, OR, AND NOT, and OR NOT.

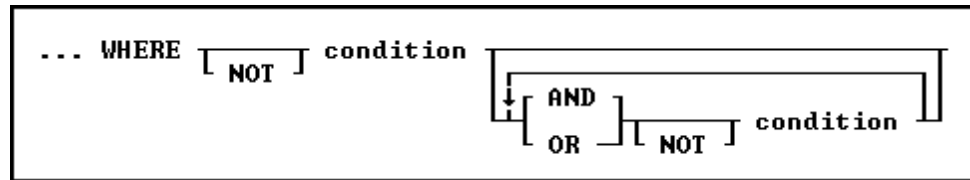
The connecting operator AND requires that both conditions it separates must be satisfied. The connecting operator OR requires that either condition it separates must be satisfied.

The connecting operator AND NOT requires that the preceding condition must be satisfied, and the following condition must not be satisfied. The connecting operator OR NOT requires that either the preceding condition must be satisfied, or any condition except the following condition must be satisfied.

In WHERE clauses with multiple conditions, conditions that are connected by AND or AND NOT are evaluated before those connected by OR or OR NOT. However, you can control the order in which conditions are evaluated by either placing parentheses around conditions or using the [SET AND](#) condition. If you set AND off, conditions are always evaluated from left to right.

WHERE Builder

When launching the WHERE Clause Builder, the following window will appear:



Examples

The following WHERE clause chooses sales amounts that are less than the value of a variable containing the daily average.

```
... WHERE amount < .dailyave
```

The following WHERE clause specifies the seventh row.

```
... WHERE COUNT = 7
```

The following WHERE clause specifies each row from the *employeetable* that contains both the first name *June* and the last name *Wilson*.

```
SELECT * FROM employee WHERE empfname = 'june' AND emplname = 'wilson'
```

The following WHERE clause selects dates in the *actdate* column that are greater than dates in the *begdate* column or are less than dates in the *enddate* column.

```
... WHERE actdate BETWEEN begdate AND enddate
```

The next three WHERE clauses use the following data:

```
empfname emplname
-----
      Mary Jones
      John Smith
      Agnes Smith
      John Brown
```

In both of the following clauses, R:BASE first evaluates the conditions connected by AND, selecting John Smith. Then R:BASE adds any Marys to the list because the connecting operator is OR. The final result includes John Smith and Mary Jones.

```
...WHERE empfname = 'Mary' OR empfname = 'John' +
      AND emplname = 'Smith'
```

```
...WHERE empfname = 'Mary' OR (empfname = 'John' +
AND emplname = 'Smith')
```

By moving the parentheses around the conditions connected by OR, you can select only John Smith. In the following WHERE clause, the first name can be either Mary or John, but the last name must be Smith.

```
...WHERE (empfname = 'Mary' OR empfname = 'John') AND +
emplname = 'Smith'
```

The following example illustrates a sub-SELECT in a WHERE clause. Assume you wanted a list of all sales representatives that had transactions greater than \$100,000, and the information for such a list was contained in two tables, *employee* and *transmaster*. The relevant columns in these tables are:

employee		transmaster	
empid	emplname	empid	netamount
102	Wilson	133	\$32,400.00
129	Hernandez	160	\$9,500.00
133	Coffin	129	\$6,400.00
165	Williams	102	\$176,000.00
166	Chou	160	\$194,750.00
167	Watson	129	\$34,125.00
160	Smith	131	\$152,250.00
131	Simpson	102	\$87,500.00
102			\$22,500.00
102			\$40,500.00
131			\$108,750.00
131			\$80,500.00
129			\$56,250.00
102			\$57,500.00
160			\$140,300.00
129			\$95,500.00
129			\$155,500.00
133			\$88,000.00
131			\$130,500.00
102			\$3,060.00
165			\$3,060.00
167			\$3,830.00
133			\$12,740.00
165			\$26,310.00

To display a list of employees in the *transmaster* table with a transaction larger than \$100,000, enter the following command:

```
SELECT empid, emplname FROM employee WHERE empid IN +
  (SELECT empid FROM transmaster WHERE netamount > 100000)
```

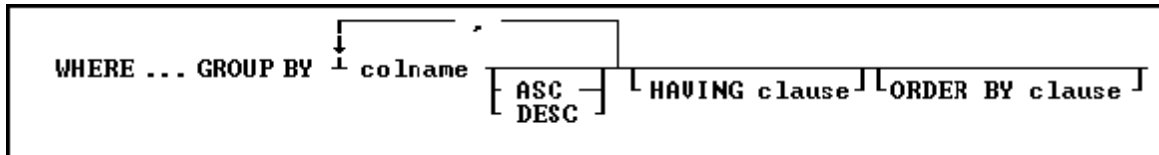
R:BASE displays the following list:

empid	emplname
102	Wilson
129	Hernandez
131	Simpson
160	Smith

Note: You can use a [sub-SELECT](#) in any command that allows a full WHERE clause.

1.23.2.1 GROUP BY

This clause determines which rows of data to include.



Options

' Indicates that this part of the command is repeatable.

ASC DESC

Specifies whether to sort a column in ascending or descending order.

colname

Specifies a column name. The column name is limited to 128 characters.

In a command, you can enter #*c*, where #*c* is the column number shown when the columns are listed with the [LIST TABLES](#) command. In an SQL command, a column name can be preceded by a table or correlation name and a period (*tblname.colname*).

GROUP BY

Returns a group of rows as a summary resulting in only unique rows. This option is generally used with [SELECT](#) commands.

HAVING clause

Limits the rows affected by the GROUP BY clause.

ORDER BY clause

Sorts rows of data.

About the GROUP BY command

This optional clause groups rows according to the values in one or more columns and sorts the results. GROUP BY consolidates the information from several rows into one row. This results in a table with one row for each value in the named column or columns and one or more values per column.

The columns listed in the GROUP BY clause are related to those listed in the command clause. Any column named in the GROUP BY clause can also be named in the command clause, but any column not named in the GROUP BY clause can be used only in the command clause if the column is used in a SELECT command.

Examples

The [SELECT](#) command clause can contain the columns named in the GROUP BY clause, and SELECT functions that refer only to columns not named in the GROUP BY clause. Because the GROUP BY clause processes information resulting from a WHERE clause, you can add a GROUP BY clause to see the sales each employee has made:

```
SELECT empid FROM transmaster WHERE netamount < $100,000 +
GROUP BY empid
```

The following intermediate result table contains columns not named in the command clause because the command clause has not been processed yet (not all the columns fit in the display, however). The first part of the processing is to group the rows by *empid*. Because seven different employees are included, the intermediate result table includes seven rows.

Intermediate Result Table-GROUP BY empid

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
4975, 4980, 5000, 5060, 5045	101, 101, 101, 101, 100	102	\$87,500, \$22,500, \$40,500, \$57,500, \$3,060
4790, 4865, 5050, 5070	104, 102, 104, 104	129	\$6,400, \$34,125, \$56,250, \$95,500
5015	103	131	\$80,500
4760, 5080, 5048	100, 100, 103	133	\$32,400, \$88,000, \$12,740
4780	105	160	\$9,500
5046, 5049	101, 102	165	\$3,060, \$26,310
5047	102	167	\$3,830

You can include more than one column in a GROUP BY clause. If you group the rows in the above example by *custid* as well as *empid*, the command looks like this:

```
SELECT empid, custid FROM transmaster +
WHERE netamount < $100,000 GROUP BY empid, custid
```

In the following table, rows are now grouped by both *empid* and *custid*, resulting in eleven groups.

Intermediate Result Table-GROUP BY empid and custid

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
5045	100	102	\$3,060
4975, 4980, 5000, 5060	101	102	\$87,500, \$22,500, \$40,500, \$57,500
4865	102	129	\$34,125
4790, 5050, 5070	104	129	\$64,000, \$56,250, \$95,500
5015	103	131	\$80,500
4760, 5080	100	133	\$32,400, \$88,000
5048	103	133	\$12,740
4780	105	160	\$9,500
5046	101	165	\$3,060
5049	102	165	\$26,310
5047	102	167	\$3,830

If one or more of the columns named in the GROUP BY clause contain null values, R:BASE forms a separate group for null values. Review the result of this [SELECT](#) command for the *employee* table:

```
SELECT empid, emplname, hiredate, emptitle FROM employee
```

<u>empid</u>	<u>emplname</u>	<u>hiredate</u>	<u>emptitle</u>
102	Wilson	03/18/90	Manager
129	Hernandez	08/28/91	Manager
131	Smith	04/14/92	-0-
133	Coffin	11/26/93	Representative
160	Simpson	01/09/94	-0-
165	Williams	07/05/92	Representative
167	Watson	07/10/92	Representative
166	Chou	07/10/93	Sales Clerk

If you group these rows by the *emptitle* column, which contains null values, you get the following intermediate result table:

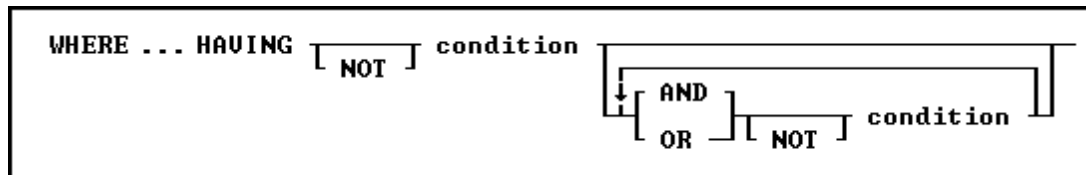
Intermediate Result Table-GROUP BY emptitle

<u>empid</u>	<u>emplname</u>	<u>hiredate</u>	<u>emptitle</u>
102, 129	Wilson, Hernandez	03/18/90, 08/28/91	Manager

133, 165, 167	Coffin, Williams,	11/26/93, 07/05/92,	Representative
	Watson	07/10/92	
166	Chou	07/10/93	Sales Clerk
131, 160	Smith, Simpson	04/14/94, 01/09/94	-0-

1.23.2.2 HAVING

This clause determines which rows of data to include based on the results of a prior [GROUP BY](#) clause.



Options

AND OR

AND indicates two conditions must both be true.

OR indicates either condition must be true.

condition

Specifies a combination of one or more expressions and/or operations that would evaluate to either true or false. See the "HAVING Conditions" below.

NOT

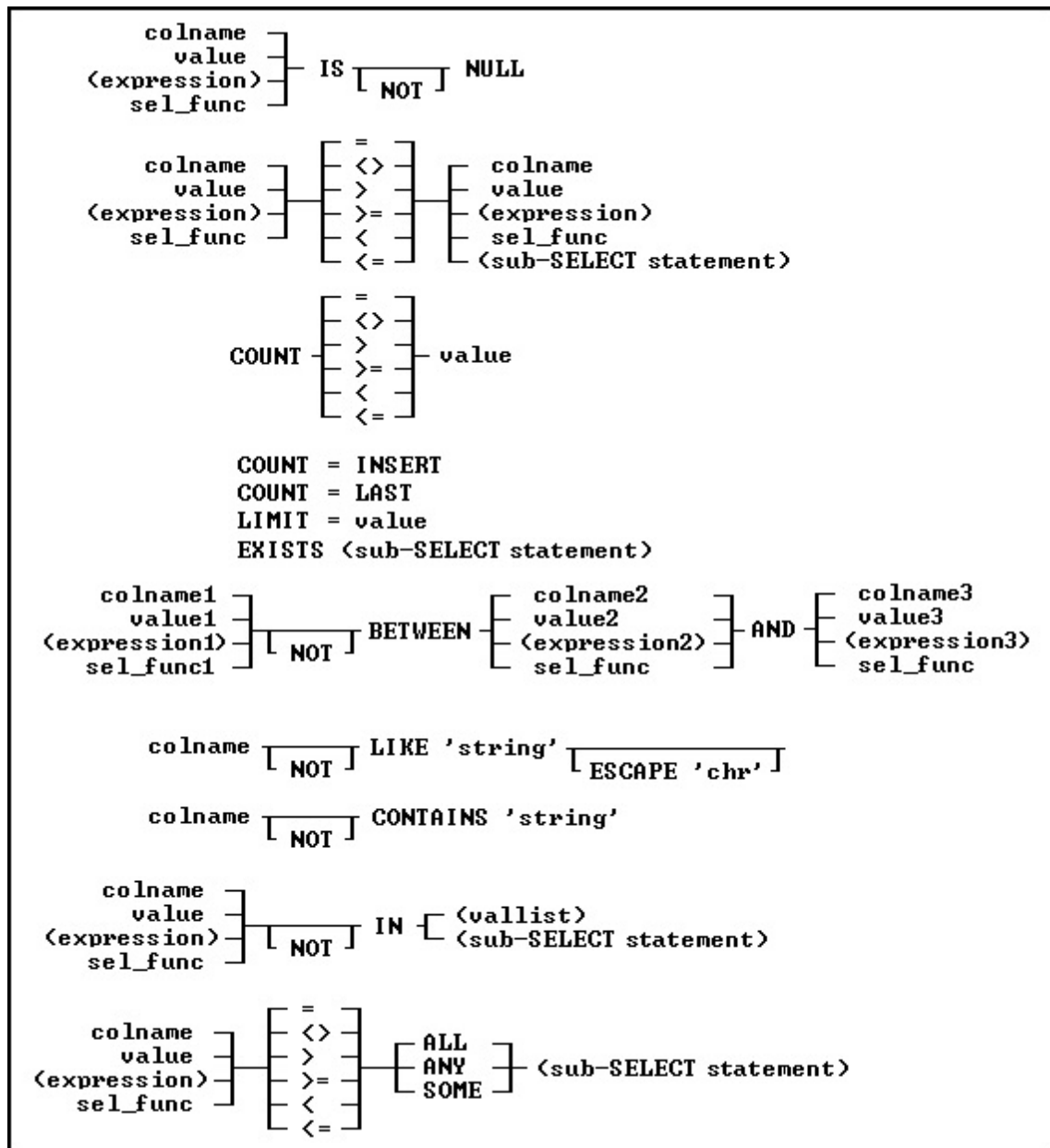
Reverses the meaning of an operator or indicates that a condition is not true.

About the HAVING command

The optional HAVING clause selects rows that meet one or more conditions from among the results of the GROUP BY clause. HAVING works the same as a WHERE clause with the following exceptions:

- A WHERE clause modifies the intermediate results of a [FROM](#) clause; a HAVING clause modifies the intermediate results of a GROUP BY clause.
- A HAVING clause can include [SELECT](#) functions.

HAVING Conditions:



Examples

To display sales information for only those employees who have made more than one sale to the same customer, add a HAVING clause such as the following to one of the examples shown previously in [GROUP BY](#). When used in a HAVING clause, [SELECT functions](#) compute results based on the values grouped in the specified column. In this HAVING clause, COUNT returns the number of values grouped in the *transid* column.

```
SELECT empid, custid FROM transmaster +
WHERE netamount < $100,000 +
GROUP BY empid, custid HAVING COUNT(transid) > 1
```

Intermediate Result Table-HAVING COUNT(transid) > 1

<u>transid</u>	<u>custid</u>	<u>empid</u>	<u>netamount</u>
----------------	---------------	--------------	------------------

4975, 4980, 5000, 5060	101	102	\$87,500, \$22,500, \$40,500, \$57,500
4790, 5050, 5070	104	129	\$6,400, \$56,250, \$95,500
5080	100	133	\$32,400, \$88,000

1.23.3 WHILE/ENDWHILE

Use the WHILE...ENDWHILE structure in a program to continuously run a set of commands based on a specified condition.

```
WHILE condlist THEN
  while-block
ENDWHILE
```

Options

condlist

Specifies a list of conditions that identify the requirements to be met.

while-block

Specifies commands to be executed if the WHILE condition is true.

About the WHILE...ENDWHILE Command

A WHILE ... ENDWHILE structure consists of conditions, commands, and an ENDWHILE statement. As long as WHILE conditions are true, R:BASE runs the commands repeatedly.

WHILE Conditions

The basic WHILE conditions are the same as those used in an [IF...ENDIF](#) structure and are as follows:

Condition	Description
<i>varname</i> IS NULL	The value of the variable is null.
<i>varname</i> IS NOT NULL	The value of the variable is not null.
<i>varname</i> CONTAINS ' <i>string</i> '	The variable has a TEXT data type and contains a ' <i>string</i> ' as a substring in the variable value. A ' <i>string</i> ' can contain wildcards.
<i>varname</i> NOT CONTAINS ' <i>string</i> '	The variable has a TEXT data type and a ' <i>string</i> ' is not contained as a substring in the variable value. A ' <i>string</i> ' can contain wildcards.
<i>varname</i> LIKE ' <i>string</i> '	The variable equals a ' <i>string</i> .' A ' <i>string</i> ' can contain wildcards.
<i>varname</i> NOT LIKE ' <i>string</i> '	The variable does not equal the ' <i>string</i> '. A ' <i>string</i> ' can contain wildcards.
<i>varname</i> LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	The variable equals a ' <i>string</i> .' If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>varname</i> NOT LIKE ' <i>string</i> ' ESCAPE ' <i>chr</i> '	The variable does not equal a ' <i>string</i> .' If you want to use a wildcard character as a text character in the string, specify the ESCAPE character <i>chr</i> . In the string, use <i>chr</i> in front of the wildcard character.
<i>varname</i> BETWEEN <i>value1</i> AND <i>value2</i>	The value of the variable is greater than or equal to <i>value1</i> and less than or equal to <i>value2</i> . The variable and the values must be the same data type.
<i>varname</i> NOT BETWEEN <i>value1</i> AND <i>value2</i>	The value of the variable is less than <i>value1</i> or greater than <i>value2</i> . The variable and the values must be the same data type.

<i>item1</i> <i>op</i> <i>item2</i>	<i>Item1</i> has the specified relationship to <i>item2</i> . <i>Item1</i> can be a column name, value, or expression; <i>item2</i> can be a column name, value, or expression.
-------------------------------------	---

The valid operators (*op*) for the conditions in a WHILE...ENDWHILE structure are listed in the table below. Do not use wildcard characters with these operators.

Operator	Description
=	Equals
<=	Less than or equal to
>=	Greater than or equal to
<	Less than
>	Greater than
<>	Not equal

A variable can be substituted for the first variable in each of the condition formats shown above, and for either item when using an operator comparison. The condition should not use dotted variables, unless the current value of that variable is to be evaluated. When comparing items with an operator, (e.g. *item1* < *item2*, *item1* >= *item2*, etc.) the condition may be enclosed in parentheses, where R:BASE will evaluate the expression each time through the loop.

You can only use wildcard characters with the LIKE, NOT LIKE, CONTAINS, or NOT CONTAINS operators. For example, *varname* LIKE '*string*%.'

You can combine conditions from the WHILE condition list by using the connecting operators AND, OR, AND NOT, and OR NOT. Be careful when using these conditions in a condition list. Conditions connected by AND are evaluated first, then conditions connected by OR are evaluated. However, you can use parentheses to set the evaluation order.

WHILE Loop Commands

All WHILE loop commands are retained in memory, so a WHILE loop runs more quickly than a [GOTO](#) or [LABEL](#) structure. A computer must have enough available memory to read all of the commands in a WHILE loop, or the program terminates abnormally.

R:BASE optimizes commands in a WHILE loop so that it runs more quickly. Use the following guidelines when constructing WHILE loops so they run more quickly.

- Do not clear variables in the WHILE loop. Rather, set those variables to null.
- Do not define variables within the WHILE loop. Only define variables outside of the loop because the values can change within the loop.
- If you issue multiple [SET VARIABLE](#) commands on a single command line, then those variables will not be optimized. If you want to increase the speed for that loop, you should put the SET VARIABLE commands on separate lines.

To turn off WHILE loop optimization, set [WHILEOPT](#) off.

The ENDWHILE Statement

ENDWHILE indicates the end of the loop. Place an ENDWHILE statement at the end of each WHILE loop. Each time R:BASE reaches the ENDWHILE statement, R:BASE returns to the WHILE command at the top of the loop and checks to see whether the conditions are still true or false. If true, R:BASE again runs the commands between the WHILE and the ENDWHILE. If false, R:BASE runs the command line immediately following the ENDWHILE.

Exiting from a WHILE Loop

To exit from a WHILE loop before the WHILE condition becomes false, use an [IF...ENDIF](#) structure to check other conditions, then use [BREAK](#) to exit from the WHILE loop. The BREAK command causes the WHILE loop to terminate when the conditions specified in the IF statement become true.

Never use GOTO to exit from a WHILE loop; use BREAK instead. BREAK clears the WHILE loop. When you do not use BREAK or the naturally occurring exit (that is, when the WHILE loop conditions are no longer true) to exit from a WHILE loop, R:BASE continues to read commands into memory. If you have a large command or procedure file, you can run out of memory and your program terminates abnormally.

Skip to the next WHILE Occurrence

Use the [CONTINUE](#) command to move to the next occurrence of the WHILE loop and run the code.

In the following example, when the code is run, processing returns to line 3 after it completes the CONTINUE command on line 6. The while-block commands in line 8 are not run.

```
SET VARIABLE v1=0
SET VARIABLE V2=1
WHILE v1 = 0 THEN
    *(while-block commands)
    IF v2 <> 0 THEN
        CONTINUE
    ENDIF
    *(while-block commands)
ENDWHILE
```

Displaying Messages

It is helpful to show a progress message when performing long running tasks in WHILE loops. A method to let the user know the status of the process is using a [PAUSE](#) 3 dialog with the [GAUGE](#) options with the [PROCESSMESSAGE](#) command, which processes messages that are currently in the windows message queue. The PROCESSMESSAGE can help in the GUI part to avoid the "Not responding" behavior in Windows operating systems.

Examples

Example 01. In the following example, R:BASE runs the commands in the WHILE block and evaluates the v2 condition in the IF statement. If v2 is not equal to zero, R:BASE runs the BREAK command and terminates the WHILE loop. R:BASE then runs the commands immediately following the ENDWHILE statement. As long as the WHILE condition (v1) is true and the IF condition (v2) remains false, the WHILE loop continues processing.

```
SET VARIABLE v1 = 0
WHILE v1 = 0 THEN
    *(while-block commands)
    IF v2 <> 0 THEN
        BREAK
    ENDIF
    *(while-block commands)
ENDWHILE
*(next command outside the while-block)
```

Example 02. The following example loops through a counter displaying a PAUSE message.

```
CLS
PAUSE 3 USING ' Calculating ... Please Stand By ...'
SET VAR vCounter INTEGER = 1
WHILE vCounter < 2500000 THEN
    SET VAR vCounter = (.vCounter + 1)
    PROCESSMESSAGE
ENDWHILE
CLEAR VARIABLE vCounter
CLS
RETURN
```


String Modifiers for All Messages

Use this symbol...	To do the following...
[<]	Left justify the data in the field.
[>]	Right justify the data in the field.
[^]	Center the data in the field.

Use this symbol...	To do the following...
[-]	Place a negative sign to the right of a negative number.
[CR]	Place "CR" to the right of a positive number.
[DB]	Place "DB" to the right of a negative number.
[()]	Enclose a negative number in parentheses.

Character Modifier	If character is a letter	If character is a number	If character is neither
_	uppercase	blank	blank
 	lowercase	blank	blank
%	uppercase	number	character
?	lowercase	number	character

Character Modifier	If character is a number	If character doesn't fill the specified format
9	number	blank
0	number	0
*	number	* if blank

If the value of `var2` is `warning`, then the above command displays:

WARNING

The following command displays the entire contents of *var3*, wrapping the text after the twelfth character. If [SET WRAP](#) is off, only the first 12 characters of text would be displayed.

```
WRITE .var3 = 12
```

The following command writes variable VARCHAR data to a text file:

```
WRITE .vVarcharData TO File.txt
```

1.24 Z

1.24.1 ZIP

Use the ZIP command to execute external programs without closing a database or leaving R:BASE.



Options

progname

Specifies the name of the external program to execute.

ROLLOUT

Executes programs that are too large to fit into the memory remaining after starting R:BASE. This option exits from R:BASE saving the current variables and environment before executing the external program. When the external program is exited, R:BASE restarts and the variables and environment are renewed. Under Windows, this will also force R:BASE to release accumulated memory that may not otherwise be released.

RETURN

This option is for R:BASE for DOS only, which causes R:BASE to return to the line in the application following the line containing the ZIP RETURN command.

C:\COMMAND.COM

This option is for R:BASE for DOS only, which starts the DOS COMMAND shell. It is best to specify the path to your COMMAND.COM, which will vary based upon the operating system version.

/C 'batchfile'

Specifies a batch file to be executed by the command shell.

About ZIP

ZIP allows you to call an external program from within R:BASE. The ZIP command is geared towards the DOS platform. For Windows, use the [LAUNCH](#) command. ZIP operates differently in R:BASE for DOS versus R:BASE for Windows. In R:BASE for Windows, you can specify a Windows program to be executed.

In either Windows or DOS you can call a batch file that uses the Windows START command to launch a Windows program. For example START /WAIT NOTEPAD in a batch file will start the Notepad program and wait for Notepad to close before returning control to R:BASE. If you do not use the /WAIT option then as soon as START had launched the Notepad program it would finish and control would pass back to R:BASE. This is because R:BASE is only watching the START. Use START /? at a command Prompt to see the other options available.

Example

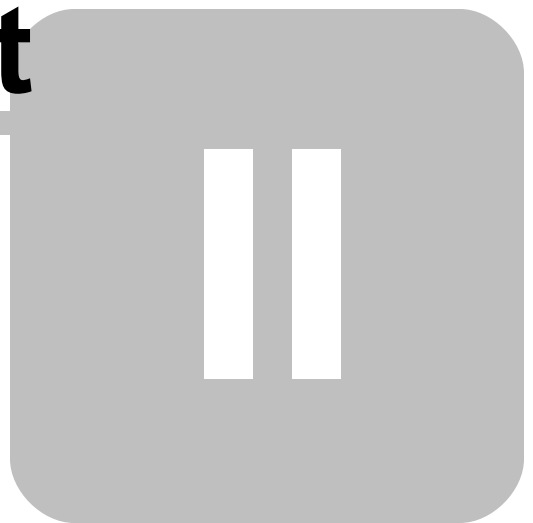
The following command runs an external program named *SmallEd*. When you exit from *SmallEd*, R:BASE continues at the point where you ran the ZIP command.

ZIP SmallEd

The following command, when run in R:BASE Windows, runs Notepad and specifies that the ABC.TXT file be opened.

ZIP Notepad ABC.TXT

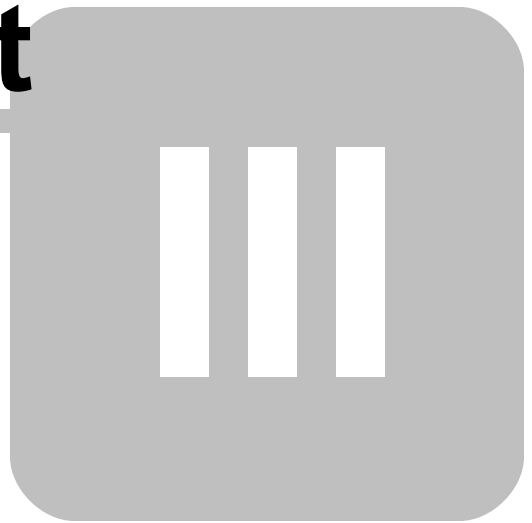
Part



2 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



3 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.

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.

Index

- - -

-- 62

- # -

#TABLEORDER 505

- { -

{ } 62

- | -

||format 101

- A -

ABORT 17
 ACCESS 172, 560
 Adjacency Factor 136
 alias 23, 40, 43, 59, 134, 443, 456, 540, 550
 AliasList 23
 ALL 560
 ALTER TABLE 17
 ampersand 512
 analytical 456
 AND 483
 ANSI 138, 308, 483
 ANSIOUTPUT 484
 APPEND 23
 appfile 60
 APPLICATION 210, 415
 Application Designer 422
 apxfile 60
 AS 473
 AS alias 132, 428
 ASCII 70, 192, 560
 ASCII text 422
 ATTACH 23, 446
 AUTOCHK 26
 AUTOCOMMIT 484

AUTOCONVERT 484
 AUTODROP 484
 AUTONUM 29, 560
 AUTORECOVER 485
 AUTOROWVER 485
 AUTOSKIP 485
 AUTOSYNC 485
 AUTOUPGRADE 486
 average 66, 87, 453
 AVG 66, 87, 453

- B -

backfile 60
 BACKUP 32
 backup file 60
 BASE64 138
 BEEP 35
 BELL 486
 bin 406
 binary 60
 binary procedure file 60
 binfile 60
 bitmap 358
 BLANK 486
 BLOB 422
 BLOB Editor 210, 415
 BLOB_EDITOR 210
 BMP 236, 358
 BOOLEAN 486
 BREAK 36, 553
 BROWSE 36
 BROWSE USING 40
 BSTR 422
 buffer 487
 build 544, 548
 Builder 326
 button settings 56
 Buttons 109

- C -

CALC 509
 calculator 121, 325
 CALL 42, 523
 CAPTION 406, 428, 486
 CASCADE 42, 284

- CASE 487, 553
- CENTURY 491
- CHARACTERS 541
- CHDIR 45
- CHDRV 46
- check 26
- CHECKPROP 487
- CHKDSK 46
- CHOOSE 47, 326
- CLBR 175
- CLEAR 57, 487, 532
- CLIPBOARD 415, 487
- CLOSE 59
- CLOSEWINDOW 59
- CLS 60
- cmdfile 60
- CMDHIST 488
- CMPAUSE 313, 489
- CODELOCK 60
- COLCHECK 489
- collate 406
- COLOR 406, 481, 490
- column 210
- Command 15, 210, 415, 428, 443, 453, 483, 484
- Command Categories 15
- command file 60
- Command History 488
- Commands 15, 17, 23, 26, 29, 32, 35, 36, 40, 42, 43, 45, 46, 47, 57, 59, 60, 62, 64, 65, 66, 68, 69, 70, 72, 75, 76, 80, 83, 87, 88, 89, 92, 93, 94, 95, 96, 122, 123, 124, 125, 128, 132, 134, 135, 136, 138, 139, 141, 142, 143, 209, 212, 213, 218, 219, 221, 223, 225, 226, 227, 230, 231, 233, 236, 237, 239, 240, 241, 243, 248, 250, 252, 258, 260, 261, 264, 265, 267, 268, 270, 271, 277, 279, 281, 282, 284, 291, 293, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 311, 313, 319, 351, 352, 353, 355, 358, 359, 361, 362, 363, 365, 370, 372, 374, 380, 382, 383, 386, 387, 389, 391, 392, 393, 399, 401, 403, 404, 406, 411, 413, 418, 419, 421, 422, 423, 424, 425, 426, 428, 429, 430, 432, 433, 435, 436, 438, 439, 440, 441, 443, 446, 448, 449, 450, 458, 459, 460, 462, 465, 467, 472, 473, 474, 476, 478, 479, 481, 482, 536, 540, 541, 544, 545, 546, 547, 548, 549, 550, 552, 553, 555, 556, 557, 558, 559, 560, 568, 571, 572, 573, 579, 581, 583, 586, 588
- COMMENT 62
- COMMENT ON 64
- COMMENTS 560
- COMMIT 65
- COMPATIB 490
- Component ID 210, 415
- COMPUTE 66
- configuration 15, 516
- CONNECT 68
- connection 88
- CONSTRAINTS 284, 560
- CONTINUE 69, 586
- Control Structures 15
- conversion 301
- CONVERT 70
- copies 406
- COPY 70, 406, 506, 540
- COUNT 66, 87, 453
- CREATE INDEX 72
- CREATE SCHEMA 75
- CREATE TABLE 76
- CREATE VIEW 80
- CREATEOBJECT 83
- CROSSTAB 87
- CSV 149, 192, 560
- CURRENCY 490
- cursor 59, 89, 139
- CURSORS 284
- CUS 192
- Custom Form Action 415
- CVAL 521

- D -

- DATA 26, 560
- Data Designer 423
- Data Dictionary 83
- Data Interchange Format 237, 359
- Data Source 443, 446
- DATABASE 321, 506, 532, 560
- Database Connect 15
- database event 304, 541
- DATE 116, 491
- DATETIME 119
- day 491
- daylight saving time 348
- DB 146, 189
- DB2 172
- dBASE 23, 284
- DBASE_TABLES 560
- DBCINN 88
- DBF 148, 190
- debug 89, 493, 516

declaration 43, 550
 DECLARE CURSOR 89, 139, 512
 decryption 308
 DEFAULT 456, 553
 Default Printer Font 308
 DEFF 92
 DELETE 93, 94, 549
 DELETE DUPLICATES 94
 DELETEOBJECT 83, 95
 DELIMIT 493, 560
 DELROW 95
 DETACH 96
 DIALOG 96, 107, 109, 110, 111, 114, 116, 117,
 119, 121, 330, 333, 341, 346
 Dialog Builder 330
 Dialog With Button Builder 333
 DIF 167, 237, 359
 DIR 122
 directory 122
 DISCONNECT 123, 304
 DISPLAY 124
 DISTINCT 36, 114, 128, 453
 DLCALL 43, 550
 DOC 162
 DROP 125
 DSN 446
 DSN-Less 446, 448, 449
 duplex 406
 duplicate 94, 506
 Duplicate Factor 136
 DUPROW 128

- E -

ECHO 494
 EDIT 128
 EDIT USING 132
 EditMask 101
 EDITOR 422, 424, 494
 EEP Specific 15
 EMAIL 282, 404
 EMF 239, 361
 emulation text file 362
 ENCRYPT 308
 encryption 308
 ENDIF 219
 ENDSW 553
 ENDSWITCH 553

ENDWHILE 583
 ENTER 134
 ENTER USING 134
 EOFCHAR 495
 EQNULL 495
 ERASE 135
 Error Handling 15
 ERROR MESSAGE 496
 ERROR MESSAGES 496
 ERROR VARIABLE 497
 ESCAPE 498
 ETXT 240, 362
 Excel 279
 EXCEPT 36, 128, 462
 EXIT 135, 304
 EXITFORM 136
 EXPLAIN 136
 EXPLODE 498
 explore 227
 EXPORT 143, 145, 146, 148, 149, 151, 153, 155,
 158, 162, 164, 166, 167, 168, 170, 172, 173, 175, 176,
 178, 179, 181, 182, 183, 185
 expression 87
 Extensible Hypertext Markup Language 393
 External Form 320
 External Form Designer 92
 external function 43, 550
 external theme 415
 ExtForm 320

- F -

FAILS 440
 FALSE 486
 FASTFK 499
 FASTLOCK 499, 511
 FCONVERT 138
 feedback 500, 593
 FETCH 139, 459
 file 122
 File Access 15
 file association 227
 file encode 138
 File Gateway 143
 FILES 500
 FILLIN 141
 find 227
 Firebird 172

FIRST 142
FIX 151
FIXED 501
fixed width 151
FKEYS 284
FLUSH 142
folder 122
FONT 406, 501
font name 308
font size 308
foreign data source 446, 449, 513
foreign database 443, 549
foreign key 136
foreign table 443
Form Designer 143
form object 83
FORMAT 491, 522
format mask 101
FORMS 143
FROM 460
function 43, 456, 550

- G -

GATEWAY 143, 145, 185, 187, 208
gauge 319
GAUGE Progress Bar 313
GET 209
GetDBVersion 321
GETPROPERTY 210
GIF 241, 363
glyph 56
GOTO 15, 212, 226
GRANT 213
Graphics Interchange Format 363
GROUP BY 474, 579

- H -

HAVING 476, 581
HEADINGS 501
HELP 218
Hex 422
HIDE 227
hour 522
HTM 153, 198
HTML 198, 243, 365, 481

hypertext markup language 365

- I -

IDQUOTES 443, 448, 501
IF 219
IF/ENDIF 219
image 422
IMPORT 143, 187, 189, 190, 192, 194, 196, 198, 199, 201, 203, 205, 206, 208
index 26, 136, 311, 502
INDEXES 560
INDEXONLY 502
INMEMORY 76, 223, 225, 284, 413, 552, 559
INNER 463
INPUT 221
input field 107
inputspec 60
INSERT 221, 502, 513, 549
integrity 26
INTERSECT 223
INTERVAL 502
INTO 459
ISTAT 512
ISV 192

- J -

JavaScript 182, 205
JOIN 225, 463
JPEG 370
JPG 248, 370
JSON 182, 205

- K -

key 344
KEYMAP 503
KEYS 311
keystroke 344
keyword 483, 541

- L -

Label 226, 424
LABELS 284
LAG 456

landscape 406
 LAST 227
 last modification 433
 LAST_MOD 433
 LAUNCH 227
 LAYOUT 503
 LBLPRINT 230, 231, 233, 236, 237, 239, 240, 241,
 243, 248, 250, 252, 258, 260, 261, 264, 265, 267, 268,
 270, 271, 277, 279, 281, 282
 LDIF 181
 LEAD 456
 LIMIT 462
 LINEEND 503
 LINES 122, 503, 512
 LIST 213, 284, 521
 LIST ACCESS 291
 list settings 55
 LISTOF 453
 LOAD 293, 509
 load directory 322
 load file 323, 324
 load folder 322
 LoadDirectoryName 322
 LoadFileName 323
 LoadFileNamePlus 324
 LOADWINDOW 210, 415
 LOCK 57, 284, 504
 log 516
 LOOKUP 114, 505
 Lotus 250, 372

- M -

MANOPT 505
 MANY 505
 MAPI 336
 MAPI Mail 336
 mask 101
 MAX 453
 MAXIMIZE 299
 MAXIMIZED 227
 maximum 66, 87, 453
 MAXTRANS 505
 MDB 203
 MDI 36, 40, 59, 128, 132, 134, 231, 353, 422, 428,
 506, 536, 540
 menu block 60
 menu file 60

menufile 60
 menuname 60
 message 107
 MESSAGES 506
 meter 313, 318
 Meter Progress Bar 313
 MICRORIM_EXPLAIN 136
 MIGRATE 300
 millisecond 522
 MIN 453
 MINIMIZE 300
 MINIMIZED 227
 minimum 66, 87, 453
 minute 522
 MIRROR 506
 MKDIR 300
 MODAL 92, 143, 424, 432
 month 491
 MOUSE 507
 MULTI 506, 507, 511
 Multi-Line 313
 MultiLine PAUSE 313
 Multi-Line PAUSE 313
 MULTISELECT 114
 MySQL 172

- N -

NAME 507
 NAMEWIDTH 508
 ndxlist 23
 Nested JOIN 463, 465
 NEWDB 301
 NEWPAGE 301
 NEWROW 301
 NEXTROW 302
 NEXTTAB 302
 no wrap 481
 NOCALC 509
 node 349
 NORMAL 227
 NORMALIZE 302
 NOScroll 481
 NOTE_PAD 510
 NOTEVARCHAR 422
 NOW 57
 NULL 302, 510
 NULLSET 302

numeric 510
NUMROUND 510

- O -

Object Manipulation 15
ODBC 446, 513, 549
ODS 206
ODT 183
offset 456
ON 282, 304, 404
ON AFTER ERROR 305
ON CONNECT 303
ON CONNECT/PACK 303
ON DISCONNECT 304
ON ERROR 305
ON EXIT 304
ONLINE 510
OPEN 306
Opening Designer Modules 15
OpenOffice 183, 206
Operating System 15
optimize 136
optimizer 136
Oracle 172
ORDER BY 307, 478
orientation 406
OUTER JOIN 465
OUTPUT 308, 484, 512
Output Devices 15
output folder 60
outputfolder 60
OVER 456
OWNER 558

- P -

PACK 311
PAGELOCK 511
PAGEMODE 512
paper name 406
paper size 406
paper source 406
Parameter 227, 415
parameters 52
PARSESEMI 512
PARTITION BY 456

PASSTHROUGH 513
PASSWORD 96, 308, 311
PAUSE 313, 337, 489
PDF 179, 252, 308, 374
PKEYS 284
PLUGIN 319, 320, 321, 322, 323, 324, 325, 326, 330, 333, 336, 337, 341, 343, 344, 346, 348, 349
PLUS 514
PNG 258, 380
popup 114, 116, 117, 119, 121
Pop-up 114
popup menu 114
Portable Network Graphics 258, 380
portrait 406
POSFIXED 514
PosgreSQL 172
precision 510
PREVROW 351
PREVTAB 351
primary key 136
PRINT 227, 352, 353, 355, 358, 359, 361, 362, 363, 365, 370, 372, 374, 380, 382, 383, 386, 387, 389, 391, 392, 393, 399, 401, 403, 404
PRINTER 233, 308, 355, 406, 514
printer name 406
PRNSETUP 406
procedure 43, 284, 515, 550
procedure file 60
PROCESSMESSAGE 411
procname 60
Program Communication 15
PROGRESS 515
Progress bar 313
PROJECT 413, 513
PROPERTY 308, 415
PSTDEV 453
PUT 418
PVARIANCE 453

- Q -

QB_WIZARD 419
QBE 419
QPRO 260, 382
QSV 192
QUALCOLS 443, 515
Qualkey 284, 443
Quattro Pro 260, 382

QUERY 136, 419
 Query Language 15
 QUIT 421
 QUOTES 415, 516, 560

- R -

R:BASE Editor 424
 R:Docs 210, 415
 R> Prompt 488
 RBA_FORM 210, 415
 RBADMIN 516
 RBAPP 422
 RBASE_FORM 210, 415
 RBASE_FORM_ACTION 415
 RBBEDIT 422
 RBDEFINE 423
 RBEDIT 424
 RBLABELS 424
 RBSYNC 425
 RBTI_ElapsedTime 500
 RBTI_PRN_EMAIL_DLG 404
 RBTI_PRNSETUP 406
 RBTI_RowsDeleted 500
 RBTI_RowsInserted 500
 RBTI_RowsUpdated 500
 RBTI_TIMEOUT 523
 RCalculator 325
 RDEBUG 516
 read-only 521
 RECALC 426
 RECOVER 428
 RECYCLE 517
 REFF 428
 REFRESH 517
 RELOAD 429
 RENAME 430
 report 210, 415, 432
 Report Designer 432
 report emulation text file 362
 ReportProperty 415
 REPORTS 432
 reserved word 483
 RESET 305, 433
 RESETROW 433
 resolution 406
 RESTORE 433
 RETURN 435, 523

REVERSE 518
 REVOKE 436
 RHELP 438
 RHIDE 438
 Rich Text 383, 422
 rich text format 383
 RICHTEXT 422
 RMDIR 438
 ROLLBACK 439
 ROSK 111
 ROWLOCKS 511, 518
 RPHONE 439
 RSHOW 440
 RSysDialog 341
 RTF 176, 261, 383, 422
 RULES 284, 440, 518, 560
 RUN 305, 441
 RUN SELECT 523
 RUNRBA 443
 RWhois 343

- S -

SATTACH 443, 446, 448, 449, 513, 515
 SAV 178
 SAVEROW 446
 Scalable Vector Graphics 264, 386
 scale 510
 schema 311, 521
 SCONNECT 443, 446, 448, 449, 513
 SCRATCH 518
 SCREEN 231, 308, 353
 screen block 60
 screen file 60
 SCREEN RESTORE 448
 screenname 60
 scrfile 60
 SCROLL 89
 SDETACH 443, 448, 449
 SDISCONNECT 443, 446, 448, 449
 search 227
 second 522
 SELECT 114, 413, 450, 453, 456, 458, 459, 460, 462, 463, 465, 467, 472, 473, 474, 476, 478, 479, 481, 512, 513
 SELECT Functions 453
 SELMARGIN 519
 SEMI 512, 519, 520

SendKeys 344
SEQUENCE 491, 522
SERVER 284, 520
SERVER_TABLES 560
SET 302, 482, 483, 484, 485, 486, 487, 488, 489,
490, 491, 493, 494, 495, 496, 497, 498, 499, 500, 501,
502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512,
513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523,
524, 525, 526, 527, 528, 529, 530, 531, 532, 536, 547
SETFOCUS 540
setting 521
share 540
SHARECPY 540
SHORTNAME 122, 520
SHOW 532, 541, 546
SHOW BUILD 544
SHOW ERROR 545
SHOW STATICVAR 546
SHOW VARIABLE 512, 536, 547
SHOW VERSION 548
SINGLE 521
SKIP 548
SLK 166
SMTP 404
SMTP Server 404
SORT 521
SORTMENU 521
source 70
speech 415
spreadsheet 399, 401
SQL 15, 172, 443
SQL Server 172
SQLite 172
SSQL 549
standard deviation 66
static variable 532, 546
STATICDB 511, 521
STATICVAR 532, 546
Status 483
STDCALL 43, 550
STDEV 66
stored procedure 523
Stored Procedures 15
STRUCTURE 26, 560
Sub-SELECT 472
SUBTRACT 552
SUCCEEDS 440
SUM 66, 87, 453
SVG 264, 386

SWITCH 553
SWITCH/ENDSW 553
SYLK 265, 387
Symbolik Link 265, 387
Syntax 15
SysDialog 341, 346
system table 571

- T -

TAB 164, 192
table 311, 415, 433, 504
table lock 57
TABLE_COLUMN 210, 415
tablealias 23
TABLEDEF 560
TABLES 560
tagged image file format 389
TALLY 555
target 70
TEMP 284
TEMPORARY 23, 76, 213, 223, 225, 284, 413, 436,
443, 552, 559
TEXT 422
text file 391
theme 52, 415
Theme Name 415
THEMENAME 111
Themes 47, 313
TIFF 267, 389
TILE 556
TIME 117, 348, 522
time zone 348
TIMEOUT 523
title 481
title settings 54
TJOURNAL 524
TOLERANCE 524
TOP 458
TRACE 524, 557
TRANSACTION 524, 525
transaction processing 15, 524, 525
TRANSPARENCY 111
tray 406
tree 349
tree view 349
TreeBuilder 349
TRIGGERS 284, 560

TRUE 486
TURBO 558
TXT 192, 268, 391
TYPE 558

- U -

UINOTIF 525
UKEYS 284
Unicode 138, 422, 527
UNICODEBSTR 422
UNION 479, 559
UNIQUE 72
UNLOAD 509, 521, 560
UPDATE 549, 568
UPGRADE 571
USER 526
USER CASE 311
UTF8 138, 308, 527
UTF-8 527
Utilities 15

- V -

Varchar 422
variable 57, 302, 536, 546, 547
Variable Handling 15
variance 66
VERIFY 527
version 320, 321, 544, 548
view 433
VIEWS 284, 560

- W -

WAIT 227, 528
Wait Until Finished 227
WALKMENU 529
WAREKI 529
WB1 260, 382
WHENEVER 572
WHERE 94, 114, 467, 573, 579, 581
WHERE Clause 52
WHERE CURRENT 89
WHILE 583
WHILE/ENDWHILE 583
WHILEOPT 529

WIDTH 512, 529
wildcard 70, 536, 546, 547
WINAUTH 530
WINBEEP 530
Window 111
Windows Metafile 392
WK1 168, 250, 372
WKS 196
WMF 270, 392
Workbook 281, 403
WQ1 170, 199
WRAP 531
WRITE 512, 586
WRITECHK 532

- X -

XHTML 271, 393
XLS 155, 158, 194, 277, 279, 399, 401
XLS 2003 401
XLS2003 279
XLSX 281, 403
XML 173, 201

- Y -

year 491

- Z -

ZERO 532
ZIP 588
zoom mode 422
ZOOMEDIT 532

Notes