

SET Characters and Keywords

- Syntax Set Special Characters:
 SET chrname=value
 chrname is one of the following: BLANK, DOLLAR, QUOTES,
 COMMA, SEMI, or PLUS.
 Set Status:
 SET keyword value
 value is one of the possible values for the associated character name or
 keyword. Table 8 provides all possible keyword values. *keyword* is one
 of the following: USER, DATE, LINES, WIDTH, ECHO, CASE,
 RULES, NULL, BELL, AUTOSKIP, REVERSE, MESSAGES,
 CLEAR, COLOR, SCRATCH, or ESCAPE.
- Purpose The SET command changes the current status of keyword settings,
 special characters, local variables, and error variables.
- Options SET SPECIAL CHARACTERS
 To change R:base special characters enter the special character name,
 an equals sign, and a new value. Table 7 shows the default values.

Table 7 Special Character Default Values

Special Character Name	Default Value
BLANK	
DOLLAR	\$
QUOTES	"
SEMI	;
PLUS	+
COMMA	,

SET STATUS

Change the current status of keyword settings by typing the keyword and its setting option. All settings except DATE are reset to default values each time you start R:base. Table 8 shows how the status keywords are used.

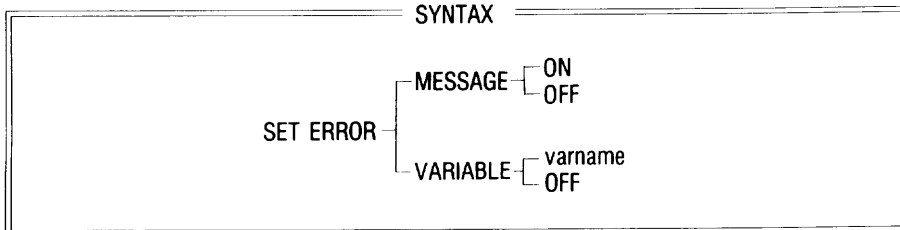
Table 8 Status Keywords Use

Keyword	Usage	Options
AUTOSKIP	autoskip to next data entry field	on or off
BELL	terminal speaker	on or off
CASE	UPPER/lower case distinction	on or off
CLEAR	clear internal buffers and transfer database files to disk after each modification	on or off
COLOR	displays menu of possible foreground and background colors available on the menu	FOREGRND color* BACKGRND color†
DATE	date format	Day-Month-Year Day-Year-Month Month-Day-Year Month-Year-Day Year-Month-Day Year-Day-Month
ECHO	echo input	on or off
ESCAPE	allows use of the [ESC] key to abort command file processing or database file access	on or off
LINES	lines per page or screen	0 (zero) or greater
MESSAGE	display diagnostic messages	on or off
NULL	null value indicator for displays	1 to 4 characters
REVERSE	reverse video in data entry fields	on or off
RULES	check data entry and modification	on or off
SCRATCH	location of temporary scratch files	on or off
USER	password	new password
WIDTH	columns per page or screen for SELECT output	39 to 132

†BACKGRND: BLACK, BLUE, BROWN, CYAN, GRAY, GREEN, MAGENTA, RED

*FOREGRND: ALL THE ABOVE, PLUS LIGHT BLACK, LIGHT BLUE, LIGHT CYAN, LIGHT GREEN, LIGHT MAGENTA, LIGHT RED, YELLOW, WHITE

SET ERROR



varname is a global variable of type INTEGER

Purpose The SET ERROR command enables programmers to write error handling routines that trap trappable R:base errors and to suppress the printing of R:base error messages.

Options SET ERROR MESSAGES OFF causes R:base to not display an error messages when an error occurs. SET ERROR MESSAGE ON causes error messages to be displayed (ON is the default).

The SET ERROR VARIABLE command defines the variable which R:base will use to hold R:base error codes. R:base sets the error variable to 0 (zero) if the command was completed without an error. A non-zero value will be set if the command execution caused an error. By checking the error variable for a non-zero value, a command file programmer can detect (or trap) many errors and execute a sequence of error-handling commands as necessary. These commands can carry out an error recovery procedure.

Comments When an error occurs in a command file, R:base normally displays a system error message. The SET ERROR options enable a command file programmer to anticipate errors and program command files so they keep running when an error occurs.

See chapter 15 in the *R:base 5000 User's Manual* for more information about global variables.

A list of the R:base error messages and their numbers is in chapter 3 of this Reference.

Examples If an error variable is defined and a trappable error occurs, R:base sets the error variable to the error code before anything else happens. It is possible to write a command file that interprets the error and prompts for additional information to correct the problem.

The syntax to set an error variable is:

```
SET ERROR VARIABLE varname
```

The syntax to return R:base to the standard error message operating mode is:

```
SET ERROR VARIABLE OFF
```

R:base requires that a database name be entered with the OPEN command (for example, OPEN *sesame*). If no database name is entered, R:base responds with system message such as Missing database name, terminates any open command files, and returns you to the R> prompt.

The following set of commands prompts the operator for a database name without terminating the command file:

```
*(SAMPLE ERROR HANDLER FOR OPEN COMMAND)
SET MESSAGES OFF
SET ERROR MESSAGES OFF
SET VAR errcnt TO 0
SET ERR VAR hold
WHILE errcnt GE 0 AND errcnt LT 10 THEN
  NEWPAGE
  FILLIN db USING "Enter database to open: " AT 4,5
  OPEN .db
  IF hold NE 0 THEN
    SET VAR errcnt TO .errcnt + 1
    IF errcnt = 10 THEN
      WRITE "Sorry, tried 10 times to open database."
      EXIT
    ELSE
      WRITE "Unable to open database, check for correct path." AT 5,5
      WRITE "Press any key to continue" AT 9,5
      PAUSE
    ENDIF
  ELSE
    BREAK          *(database opened, leave WHILE loop)
  ENDIF
ENDWHILE
```

SET POINTER

SYNTAX

```
SET POINTER #n [varname] FOR tblname [SORTED BY collist] [WHERE condlist]
```

- Purpose** The SET POINTER command sets up a route that quickly moves a pointer (1, 2, or 3) through the rows in a table, and sets the pointer to the first row in the route. Each time a NEXT command for that pointer is executed, the pointer is moved to the next row in the established route (see the NEXT command description and the example command file below).
- Options** SET POINTER without a SORTED BY option or WHERE option sets the route to all of the rows in the table in the order in which the rows were added to the table.
- varname*: This variable is set to 2406 when there are no more rows to point to or when there are no rows in the table. If there is a row to point to next, this variable is set to zero.
- SORTED BY...*: When this option is used, the NEXT command moves the pointer from row to row in the sort order.
- WHERE...*: With this clause only the rows which have values that meet the WHERE clause condition are included in the route.
- Comments** The R:base SET VAR, CHANGE, ASSIGN, and DELETE commands can make use of the current row in a route by using the #n form of those commands as shown in table 9.

Table 9 Using a Route Number in a Command

What to do...	How to do it...
Set a variable to a column value.	SET VAR varname TO colname IN #n
Change a column value	CHANGE colname TO value IN #n CHANGE colname TO varname IN #n
Assign the value of an expression to a column	ASSIGN colname TO expression IN #n
Delete the row	DELETE ROW FROM #n

The three R:base route numbers can be active at the same time. They can point into the same table or different tables.

Examples To step through a table, the NEXT command is used inside a WHILE loop, as shown in the following example.

Assume you have a four-row table called *oldcust* in the database *compuco* which holds the following values:

<i>CUSTID</i>	<i>COMPANY</i>	<i>YTDPURCH</i>
100	PC Distribution Inc.	\$100,000.00
101	Computer Distributors Inc.	\$75,000.00
102	Industrial Computers Inc.	\$95,000.00
103	Computer Mountain Inc.	\$65,000.00

```
OPEN compuco
SET POINTER #1 status FOR oldcust ; SHOW VAR status
SET MESSAGES OFF
  WHILE status = 0 THEN
    SET VAR vcustid TO custid IN #1
    SHOW VAR vcustid
    NEXT #1 status ; SHOW VAR status
  ENDWHILE
QUIT
```

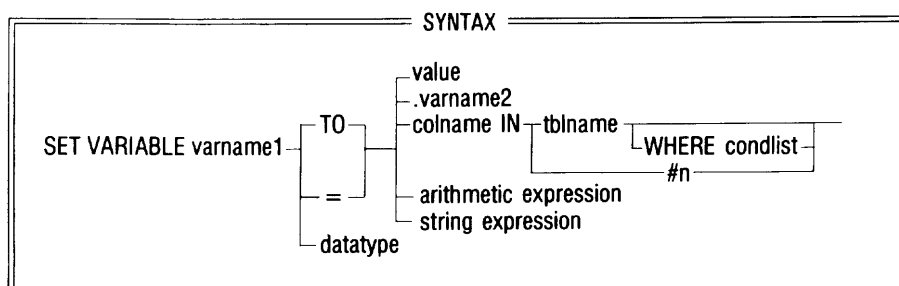
This command file does the following:

1. Opens the database.
2. Sets up the route for moving the #1 row pointer through the table *oldcust*.
3. Sets messages off so diagnostic and informational messages do not appear on the screen.
4. Sets up a WHILE loop that allows each row to be pointed to in order.
5. Within the loop for each row:
 - The value of column *custid* is moved into variable *vcustid*
 - The value of *vcustid* is displayed
 - The row pointer is advanced
 - The variable *status* is displayed to show whether there was another row to point to.

Following is the output when the program is run:

0
100
0
101
0
102
0
103
2406

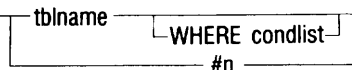
SET VARIABLE



Purpose The SET VARIABLE command defines or redefines a variable value and/or datatype.

Options The SET VARIABLE command syntax shows six options:

- (1) SET VAR varname1 datatype
- (2) SET VAR varname1 TO value
- (3) SET VAR varname1 TO .varname2
- (4) SET VAR varname1 TO arithmetic expression
- (5) SET VAR varname1 TO string expression
- (6) SET VAR varname1 TO colname IN



Comments (1) SET VAR varname1 datatype

varname1 is the name of variable being set.

datatype refers to one of the valid R:base data types: INTEGER, REAL, DOLLAR, TEXT, DATE, or TIME.

For a new variable, this option defines a variable, sets its data type, and sets its value to null. Use this form of the SET command to establish the variable data type before the variable is used to accept data. For an existing variable, this option can be used to change the data type.

(2) SET VAR varname1 TO value

varname1 refers to the name of the variable being set.

value is a data value, such as 10, TOM, 3.1416, or \$17.23.

If *varname1* already exists, any new *value* must be the same data type as *varname1*. If *varname* defines a new variable, R:base assigns the variable the data type based on the *value*.

(3) SET VAR varname1 TO .varname2

varname1 refers to the name of the variable being set.

varname2 is the name of an existing variable. The period preceding *varname2* directs R:base to set *varname1* to the value of *varname2*.

(4) SET VAR varname1 TO arithmetic expression

varname1 refers to the name of the variable being set.

arithmetic expression is an arithmetic operation (+, -, x, /, or %) which combines two items. The items may be assigned values or the values of variables.

If *varname1* is not defined, the data type of each item determines the data type of the result. For example, if an INTEGER is added to a REAL, the resulting variable is REAL unless the result variable has been explicitly typed as INTEGER.

Refer to chapter 15 of the *R:base 5000 User's Manual* for a summary of the data types R:base assigns to expressions of the form:

result = v1 op v2

Note that if either of the items in an expression is null, the result will always be null.

(5) SET VAR varname1 TO string expression

varname1 refers to the name of the variable being set.

string expression is a concatenation (+ or &) which combines two items. The items may be string values or dotted variables of any type.

Following are the R:base string operations:

- + Combines two items without a space between them
- & Combines two items with a space between the two

As an example of the expression *item1* + *item2*, if *item1* is MAN, and *item2* is AGEMENT, the resulting variable value would be MANAGEMENT.

The expression *item1* & *item2* also connects *item2* to *item1*, but leaves a space between the two items. For instance, if *item1* is FRANCES and *item2* is RICHARDS, the resulting value is FRANCES RICHARDS.

If either item in the expression is null, the result will be null.

(6) SET VAR varname1 TO colname IN

tblname	WHERE condlist
#n	

varname1 refers to the name of the variable being set.

colname and *tblname* refer to a specific column and table in the open database.

condition in the WHERE clause specifies the row that holds the value.

#n is a route number (1, 2, or 3) that points to a specific row.

If the optional WHERE clause is not used to specify a row when a table name is used, the value for the column in the first row is used.

A route pointer set by the SET POINTER command can be used in lieu of a table name and WHERE clause.

A dotted variable may be used in place of the column name. For example,

```
FILLIN v1 USING "Which column do you want to update? "  
SET VAR newpr TO .v1 IN product
```

In this way, you can use the same SET VAR command to access different columns in a table.

The number of variables available depends on the amount of memory in the computer and the length of the data items that are in the variables. At least 40 variables can be defined at any one time in the global variable area and in some cases over 1,000 variables may be defined, depending on the amount of RAM you have available.

Variables are created, or defined, by using one of the following commands:

```
SET VARIABLE varname TO  
SET ERROR VAR varname  
SET POINTER #n varname FOR tblname  
FILLIN  
DRAW  
COMPUTE  
EDIT VARIABLE  
ENTER VARIABLE
```

After being defined, a variable name can be used in IF and WHILE conditions (see IF...THEN...ENDIF and WHILE...ENDWHILE in this reference).

A variable name may also be used in any R:base command when it is prefaced by a period (or dot); an example of such a dotted variable is *.v1*. The occurrence of a dotted variable in an R:base command causes the current value of the variable to be used in the command. For example, the commands

```
SET VARIABLE v1 TO "sales"
```

```
SELECT ALL FROM employee WHERE dept EQ .v1
```

select the rows in the *employee* table where the value of the column *dept* is equal to the value in the variable *v1*, or in this case *sales*.

Use the SHOW VARIABLES command to display the current value and data type of all the currently defined global variables. Use the SHOW VAR *varname* command to display the current value of one.

The command

```
SET ERROR VAR ev1
```

sets the data type of variable *ev1* to INTEGER.

The command

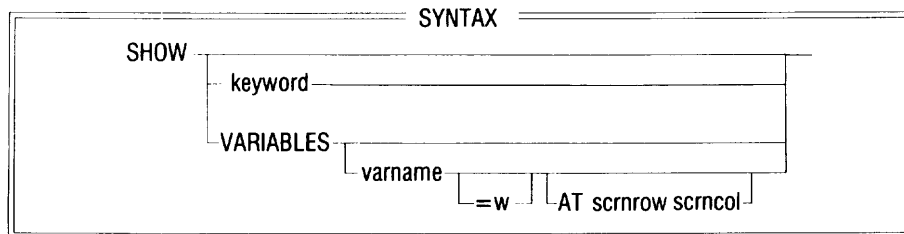
```
SET POINTER #1 e1 FOR table1
```

sets the data type of *e1* to INTEGER and its value to 0.

Examples

See above and chapter 15 of the *R:base 5000 User's Manual*.

SHOW



- Purpose** The SHOW command directs R:base to display the status of system parameters, and all format and default conditions defined with the SET command. The SHOW command also displays the status of the buffer clearing function, current user identification, and global variable values.
- Options** *keyword* is one of the following: AUTOSKIP, BELL, CASE, CLEAR, DATE, ECHO, ESCAPE, LINES, MESSAGES, NULL, REVERSE, RULES, USER, WIDTH
- VARIABLES* displays all the global variables in the database.
- varname* displays the variable *varname*.
- =w* sets the maximum column width for the display.
- AT scrnrow scrncol* defines the screen row and column where the value is displayed.
- Comments** To display all system characters and keywords (except CLEAR, ESCAPE, RULES, and USER), type:
- SHOW

Examples **SHOW CLEAR**
 Displays buffer clear status.

SHOW RULES
 Displays the rules defined for the database and the status of the rule
 checking function.

SHOW USER
 Displays the current USER password.

SHOW VARIABLE totalsales AT 10,25
 Displays the value of the variable *totalsales* beginning at line 10 and
 column 25 on the screen.

SHOW ERROR

SYNTAX

SHOW ERROR varname AT scrnrow scrncol

varname: Refers to the variable that has been set up by the execution of a SET ERROR VARIABLE command to be the variable which R:base sets to the numeric code of a system error message before that message is displayed

AT scrnrow scrncol: Defines the screen row and column at which to display the first character of the variable contents

Purpose: The SHOW ERROR command displays the error code value of the R:base system error message that is to be displayed.

SHOW VARIABLES

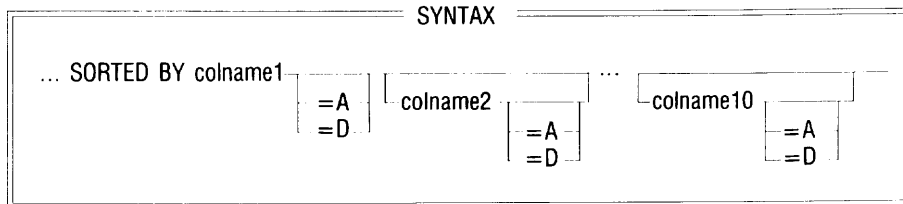
SYNTAX

```
SHOW VARIABLES [varname] [=w] [AT scrnrow scrncol]
```

w is the display width

- | | |
|----------|--|
| Purpose | The SHOW VARIABLES command displays the current values of all the global variables or the value of one particular global variable. |
| Options | <p>The SHOW VARIABLE command without options will display all the global variables.</p> <p><i>varname</i>: Specifies a particular variable to be displayed.</p> <p><i>=w</i>: Specifies a maximum width for the display of the variable. Long text variables are displayed on multiple lines.</p> <p><i>AT scrnrow scrncol</i>: This clause defines the screen row and column at which to display the first character of the variable contents field.</p> |
| Comments | <p>SHOW VARIABLES is useful for including variable values in informational displays for an application operator.</p> <p>SHOW VARIABLES can also be used for command file debugging purposes, to trace the changing values of one or more variables (see chapter 15 in the <i>R:base 5000 User's Manual</i> for more information).</p> |
| Examples | <p>SHOW VARIABLES</p> <p>Displays all the global variables.</p> <p>SHOW VAR status AT 10 15</p> <p>Displays the contents of the variable <i>status</i> starting at screen row position 10 and screen column position 15.</p> <p>SHOW VAR addr2=22 AT 10 15</p> <p>Displays the contents of the variable <i>addr2</i> starting at screen row position 10 and screen column position 15. Assuming, for purposes of example, that <i>addr2</i> has data type TEXT with length 30, the contents will be displayed on three lines, each 10 characters wide.</p> |

SORTED BY Clause



- Purpose** The sort clause is used with several R:base commands to specify the order in which rows of data from a table are displayed.
- Options**
- =A: Sorts in ascending order. Follows the column name to be sorted in ascending order. =A is the default value, and does not need to be entered.
 - =D: Sorts in descending order. Follows the column name to be sorted in descending order.
- Comments** Several commands allow you to use the optional SORT clause feature. The syntax for the SORT clause is the same regardless of the command.
- You can specify up to 10 columns in the clause. Sorting is performed in ascending order if you append =A to the column name, or in descending order if you append =D to the column name. If nothing is appended to the column name, the column is sorted in ascending order.
- Examples**
- SELECT ALL FROM transx SORTED BY custid prodid
- Sends all columns in table *transx* to the current output device sorted by *custid* with a secondary sort by *prodid*.
- EDIT prodid proddesc location FROM product SORTED BY prodid=D
- The values for the specified columns are displayed in descending product ID order on the screen for editing.

SUBTRACT

SYNTAX

```
SUBTRACT tblname1 FROM tblname2 FORMING tblname3
                                USING collist
```

tblname1 is an existing table

tblname2 is an existing table

tblname3 is the table to be formed

collist is one or more columns in *tblname2*

- Purpose** SUBTRACT is a relational command that forms a new table made up of rows from a table (*tblname2*) that do not match those of another (*tblname1*).
- Comments** R:base looks in each table for at least one column that has the same name as a column in the other table. R:base compares common columns from each row of the first table with corresponding columns from every row of the second table. Each row of the second table that has no match in the first table is copied to the new table.
- Options** *USING...*: This clause lets you list one or more columns (*collist*) for the new table from the second existing table. You can also use the clause to arrange the order of the columns in the new table. If you do not include the clause, R:base uses all columns in *tblname2* when creating *tblname3*.

Example SUBTRACT pay84 FROM emptable FORMING newtable

Figure 5 shows the results of the command.

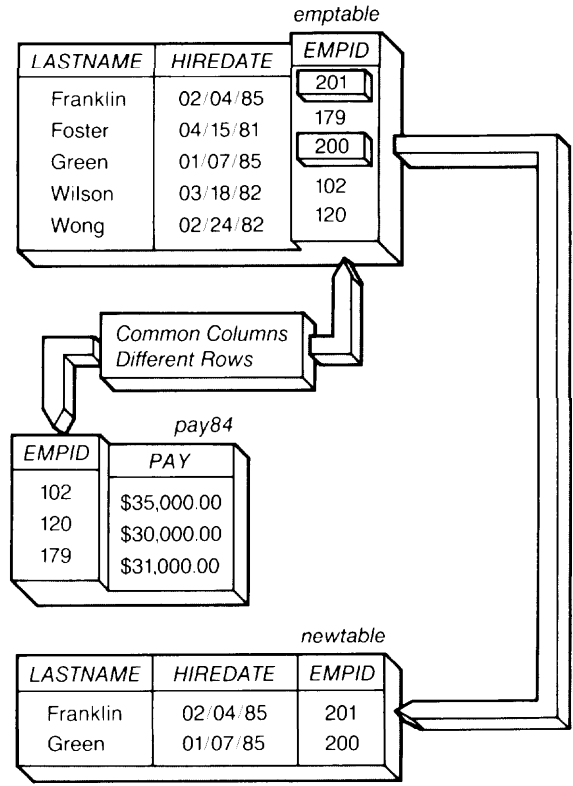


Figure 5 An example of the SUBTRACT Command

TABLES

SYNTAX

TABLES
tblname WITH collist

- Purpose** The TABLES command starts the table definition process in the DEFINE mode. After you enter the TABLES command, enter the table name and list the columns that it contains.
- Comments** After you enter the TABLES command, you can enter as many tables as you need.
- The columns you specify for the tables must already be defined. See the COLUMNS command for additional information.
- If there is an owner password already specified for the database, use the USER or OWNER commands to supply this password so you may modify the database definition.
- When you define tables, the order in which you specify the columns is the order that R:base follows for loading data. One table can have a maximum of 400 columns and each database can have a maximum of 40 tables. The maximum number of columns that each database can handle is also 400.
- Example** TABLES
transx WITH transid empid custid prodid units price tdate
- Defines a table named *transx* that consists of the columns *transid*, *empid*, *custid*, *prodid*, *units*, *price*, and *tdate*.

TALLY

SYNTAX

```
TALLY colname FROM tblname [WHERE condlist]
```

- Purpose** The TALLY command counts how many values in a column are identical. Then, it displays each unique value and how many times it occurs.
- Options** *WHERE...*: This clause specifies which rows to include in the tally.
- Comments** The results of the tally are displayed as a table.
- Examples** TALLY custid FROM transx
Lists the customer identification numbers in table *transx* with a count of the number of occurrences of each. R:base displays the information in ascending sorted order.

TYPE**SYNTAX**

TYPE filespec

Purpose	The TYPE command displays the contents of an ASCII text file.
Comments	<p>This command is similar to the DOS TYPE command. You can use it when R:base is running for the same functions and in the same ways that you would use it elsewhere under DOS.</p> <p>If used within R:base, the TYPE command is most often used to display menu text, a multi-line operator prompt, a help screen, or any ASCII file. Note, however, that the R:base DISPLAY command can perform these same functions. DISPLAY is the recommended command to use in command files to display text, because DISPLAY can display both ASCII text files and non-ASCII text blocks that have been put into procedure files by RCOMPILE. The TYPE command can only display ASCII text files.</p> <p>Another use of the TYPE command is to display the contents of ASCII command files in order to review the contents—for example, to refresh your memory as to which variables are used by a command file or to read the comments embedded in a command file. If you intend to change the command file, use RBEDIT both to display and to change the contents of an ASCII command file.</p> <p>See your DOS manual for information on messages generated by the TYPE command.</p>
Examples	<p>TYPE b:\mydir\myfile.dat</p> <p>Executing this command displays the ASCII file <i>myfile.dat</i> stored on drive <i>b:</i> in directory <i>mydir</i>.</p> <p>TYPE thisfile.dat</p> <p>Executing this command displays the ASCII file <i>thisfile.dat</i> stored on the default drive and current directory.</p>

UNION

SYNTAX

```
UNION tblname1 WITH tblname2 FORMING tblname3
      USING collist
```

tblname1 is an existing table

tblname2 is an existing table

tblname3 is a table to be formed from *tblname1* and *tblname2*

collist is one or more columns in either *tblname1* or *tblname2* or in both

- | | |
|----------|---|
| Purpose | The UNION command combines the columns and rows of two existing tables (<i>tblname1</i> and <i>tblname2</i>) in a new table. R:base looks in each table for at least one column that has the same name as one in the other table. When R:base finds all common columns, it builds a new table that contains all columns from the two tables, combining rows with identical values in each column into a single row. |
| Comments | <p>If values for the common columns are found in a row from each table, R:base combines the two rows into one and copies the combined row to a new table. If the value is found in a row from only one table, R:base copies that row and completes it with null values for the columns of the other table.</p> <p>The order in which the tables are specified in the command determines the order of the columns in the new table. Columns from <i>tblname1</i> are placed first, and columns from <i>tblname2</i> that are not common columns are placed after those from <i>tblname1</i>.</p> |
| Options | <i>USING...</i> : This clause lets you list one or more columns (<i>collist</i>) from either one of the existing tables or from both. At least one of the columns in your 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. |

Example UNION emptable WITH pay84 FORMING newtable

Figure 6 shows the results of this example.

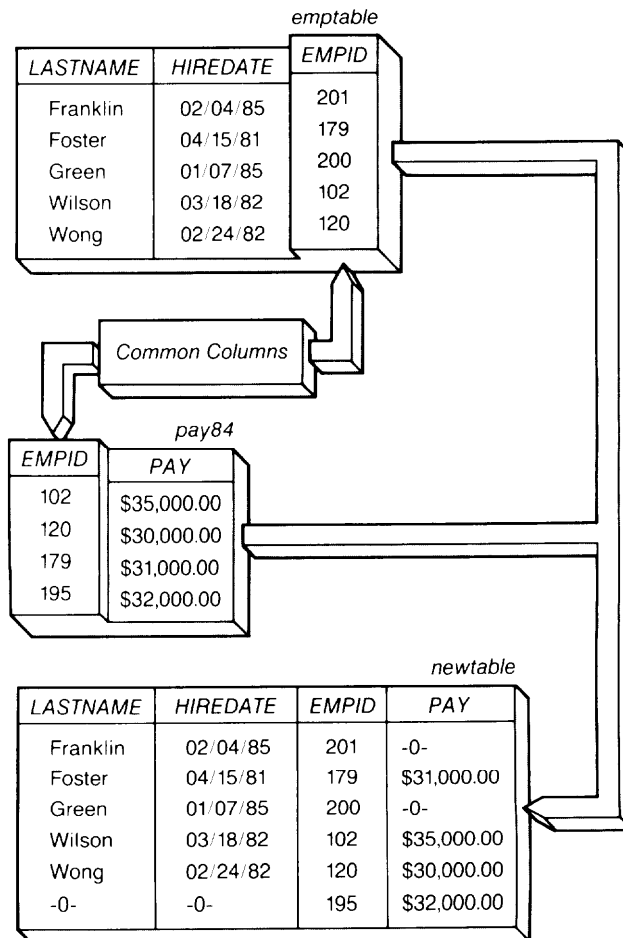
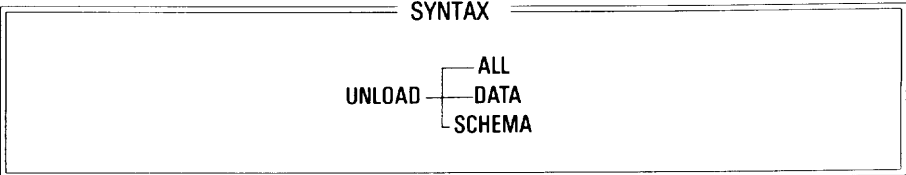


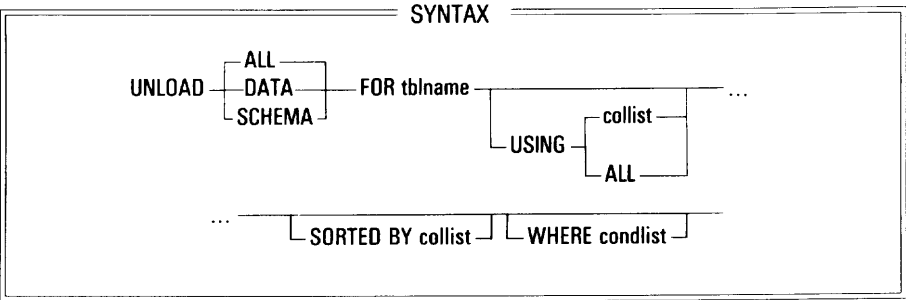
Figure 6 An Example the UNION Command

UNLOAD

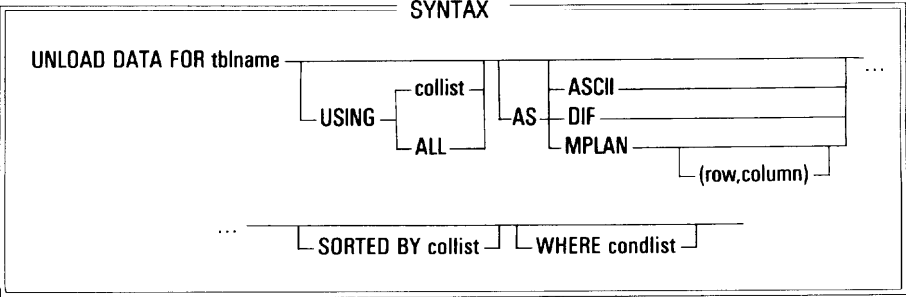
To unload all tables using all columns, use this form:



To unload a specific table with selected or ALL columns, use this form:



To unload a specific table in a specific file format with selected or ALL columns, use this form:



(row,col) is the Multiplan coordinate specification

Purpose	The UNLOAD command is used to convert data and definitions into formats which can be processed by other programs or machines or is used to transfer tables from one database to another.
Options	<p><i>UNLOAD ALL, UNLOAD DATA, UNLOAD SCHEMA</i>: Allows you to specify the part of the table to unload. ALL indicates unloading both data and table structure. DATA indicates unloading data only. SCHEMA indicates unloading the table or database structure only.</p> <p><i>FOR tblname</i>: Specifies a single table to be UNLOADED. If this clause is not used, then all tables in the database are UNLOADED.</p> <p><i>USING...</i>: Specifies which columns and in what order you want the columns UNLOADED. ALL means to use all columns in the order they are defined in the table.</p> <p><i>AS ASCII, AS DIF, AS MPLAN</i>: Allows you to specify the output file type—ASCII delimited, DIF, or Multiplan SYLK. Use the AS... clause only if you are unloading DATA. Omit the AS... clause if unloading SCHEMA or ALL.</p> <p><i>SORTED BY...</i>: This clause sorts the unloaded data by specific columns. This clause can only be used if <i>tblname</i> is specified.</p> <p><i>WHERE...</i>: This clause selects specific rows to be unloaded. This clause can only be used if <i>tblname</i> is specified.</p>

Comments UNLOAD can transfer any of the following combinations:

- The structure and data of every table in a database
- The structure of every table in a database
- The data for every table in a database
- The structure and data of an individual table
- The structure of an individual table
- The data for an individual table

If the table name is omitted the unload process is applied to all tables in the database. If any of the AS... clauses are used, however, a table name must be included.

An OUTPUT command must precede the UNLOAD command to direct the unloaded data to the printer or to a file. The file, in turn, may be used in subsequent R:base operations or by other programs such as VisiCalc (DIF files), Multiplan (SYLK files), or any program which accepts ASCII delimited format.

ASCII delimited files are characterized by separation of individual values with a comma or space used as a delimiting character. Text values with embedded blanks must be surrounded with quotation marks. For example, "This is text entry",123,\$25.00,"more text entry"... .

The columns unloaded may be limited by the USING clause. The rows selected for unloading may be limited by the WHERE clause. The data can also be sorted with the SORTED BY clause.

UNLOADED information can be transferred back into R:base using one of two methods. The INPUT command is used to load information that has been generated using any UNLOAD syntax other than UNLOAD...AS... If the AS clause was used to unload, the data must be loaded back into R:base using the LOAD command.

If the UNLOAD SCHEMA format is used, a copy of the table structure can be INPUT without the data after the database name is changed in the unloaded file. If you use the UNLOAD SCHEMA or UNLOAD ALL form, UNLOAD outputs a DEFINE *dbname* command as the first line in the output file. If you need to load the data or schema into a different database, you can change the database name on the DEFINE line using an editor.

Examples `OUTPUT myfile.dbs`
 `UNLOAD ALL`

The entire structure and data for every table in the currently open database is unloaded to the file named *myfile.dbs*. When the UNLOAD process is complete, you should redirect output back to the screen with the `OUTPUT SCREEN` command.

`UNLOAD DATA FOR product AS ASCII`

Unloads only the data as an ASCII delimited file. Places quotes around each field and separate fields with commas.

`UNLOAD ALL FOR empfile USING empid frstname lastname SORTED BY + empid`

Unloads data and schema for table *empfile* but limits the columns to those listed, sorting the rows by *empid*.

`UNLOAD SCHEMA FOR custlist`

Unloads the structure for table *custlist*. An `INPUT` command builds a structure identical to the structure of the table in the first database. If you want the table created in a database with a different name, the unloaded file must be edited to change the database name on the `DEFINE` line.

USER

SYNTAX

USER password

password is the code (OWNER, MPW, or RPW) that restricts access to a database or table

- Purpose The USER command operates with the PASSWORDS command to regulate access to the database. When you enter the USER command, R:base identifies your level of access to the database.
- Comments The password you specify in the USER command allows you to modify the data in a table (MPW), to read the data in a table (RPW) or both. If the (OWNER) password is specified with the USER command, the operator can also modify the database structure.

USING Clause

SYNTAX

```
... USING collist
... USING formname
... USING "message"
```

Purpose	The USING clause specifies a list of column names to be included in the process dictated by the command preceding the clause.
Options	<p><i>collist</i> specifies a list of columns.</p> <p><i>formname</i> specifies a table form name used with the ENTER or EDIT commands.</p> <p><i>message</i> specifies a prompt message to be used with FILLIN.</p>
Comments	<p>The USING clause can be used with these commands: EDIT USING formname, EDIT VARIABLES, ENTER VARIABLES, FILLIN, LOAD, INTERSECT, JOIN, PROJECT, RUN, SUBTRACT, UNION, and UNLOAD.</p> <p>When used with JOIN, USING indicates which columns to use as the basis for joining two tables.</p> <p>With other commands (INTERSECT, PROJECT, SUBTRACT, and UNION), the column list indicates the columns for the new table.</p>
Examples	<p>UNLOAD DATA FROM transx USING transid custid</p> <p>Unloads the data values of columns <i>transid</i> and <i>custid</i> only from table <i>transx</i>. No other columns are unloaded.</p> <p>EDIT USING tranform</p> <p>Edits the table associated with the table form <i>tranform</i>.</p> <p>JOIN empfile USING empid WITH salesrep USING empid FORMING newtable</p> <p>Creates <i>newtable</i> by joining the tables <i>empfile</i> and <i>salesrep</i> based on matching employee numbers in column <i>empid</i>.</p>

WHERE Clause

SYNTAX

```
... WHERE condition1
        [AND condition2] [AND condition10]
        [OR condition2]  [OR condition10]
```

condition1 is the first condition

- | | |
|----------|--|
| Purpose | The optional WHERE clause is used in several R:base commands to qualify or to restrict the rows affected by a command. |
| Options | <p>AND condition: AND is a logical operator which indicates that the specified condition must be met along with the primary condition. See Comments for a discussion of logical operators.</p> <p>OR condition: OR is a logical operator which indicates that the specified condition can be met in place of the primary condition. See Comments for a discussion of logical operators.</p> |
| Comments | <p>Use the same syntax in all commands except JOIN (see JOIN).</p> <p>When more than one condition in a WHERE clause is used, the clauses are connected using the operands AND and OR. Up to ten comparisons can be included in one command. AND requires that the two conditions it separates must both be satisfied. OR stipulates that any of the conditions it separates can be satisfied.</p> <p>R:base evaluates WHERE conditions from left to right within the clause. Use these forms for WHERE conditions:</p> <ul style="list-style-type: none"> ■ <i>colname EXISTS</i>: True if the value of the column is other than NULL ■ <i>colname FAILS</i>: True if the value of the column is NULL (missing) ■ <i>colname OP value</i>: Where <i>OP</i> is EQ, NE, GT, GE, LT, or LE ■ <i>colname OP colname</i>: Where <i>OP</i> is EQA, NEA, GTA, GEA, LTA, or LEA ■ <i>colname CONTAINS string</i>: True if the column contains the string ■ <i>LIMIT OP value</i>: Where <i>value</i> is integer and <i>OP</i> is EQ, or GE ■ <i>COUNT OP value</i>: Where <i>value</i> is integer and <i>OP</i> is EQ, or GE ■ <i>COUNT OP LAST</i>: Where <i>OP</i> is EQ or GE and LAST refers to the last row in the table |

You can use the wildcard character * when comparing a column to a text value. For example,

```
... WHERE lastname EQ "B*"
```

selects rows where the column *lastname* starts with the letter B.

If you compare a column with a value, you can enter the value or you can specify a global variable instead. If you specify a variable, R:base compares the column with the most current value of the variable. Precede the variable name with a period when you enter the variable name.

The LIMIT EQ clause lets you specify the number of rows the command affects. It does not count as one of the ten WHERE conditions if it is specified last. For example, you can use up to ten WHERE conditions followed by: AND LIMIT EQ value.

You can significantly reduce the time R:base takes to process a WHERE clause by using KEY processing. Time may be reduced if all three of the following conditions exist:

- The last condition in the clause must refer to a key column
- The last operator used in the condition must be EQ or =
- If more than one condition is specified in the clause, the last condition must be preceded by the logical operand AND

Examples

```
WHERE amount LT .dailyave
```

Chooses sales amounts that are less than a variable that contains the daily average.

```
WHERE COUNT EQ 7
```

The COUNT EQ operator lets you specify the particular row affected by the command. This clause specifies the seventh row.

```
WHERE COUNT EQ LAST
```

Specifies the last row of a table.

```
SELECT ALL FROM empfile WHERE frstname EQ june AND  
lastname EQ wilson
```

Suppose your database stores first names in one column and last names in another. If you want to search the table *empfile* for the row that has information about June Wilson, require that the first name *and* the last name exist in the same row.

WHILE...THEN...(BREAK)...ENDWHILE

SYNTAX

```
WHILE condlist THEN
    while-block
ENDWHILE
```

or

SYNTAX

```
WHILE condlist 1 THEN
    while-block
    IF condlist THEN
        BREAK
    ENDIF
ENDWHILE
```

condition is a phrase that evaluates true or false conditions. The *condition* can contain up to ten simple conditions combined with the operands AND and OR.

while-block is one or more R:base commands that are executed as long as the *condition* is true.

Purpose The WHILE command enables looping structures in R:base command files.

Options IF condition2 THEN
BREAK
ENDIF

is an optional function within a WHILE command. The BREAK command causes the WHILE loop to terminate when the *condition 2* specified in the IF statement becomes true.

Comments As long as the WHILE *condition* is true, R:base repeatedly executes the commands between the WHILE ... THEN statement and its corresponding ENDWHILE statement.

 The ENDWHILE ends the loop. Each time ENDWHILE executes, control is returned to the corresponding WHILE command at the top of the loop and *condition* is checked to see if it is true or false. If false, the next command line executed is the one that immediately follows the ENDWHILE. If true, the commands between the THEN and its matching ENDWHILE are executed again.

 Up to ten simple conditions may be combined in the WHILE *condition* by using the operands AND and OR. The five kinds of conditions are the same conditions used in an IF command, and are shown in table 11.

Table 11 WHILE Conditions

Kind of Simple Condition	Expression evaluates to true if...
varname EXISTS	The value of the variable is other than NULL.
varname FAILS	The value of the variable is NULL.
varname CONTAINS string*	The variable is of data type TEXT, and <i>string</i> is contained as a substring within the text string stored in the variable.
varname op value (op is EQ, NE, GT, GE, LT, or LE)	The value of the variable has the specified relationship (equal, not equal, etc.) to the value in the condition.
varname1 op .varname2 op is EQ, NE, GT, GE, LT, LE, =, < >, >, > =, <, or < =	The values of the two variables have the specified relationship (equal, not equal, etc.).
*You can use NE *string* to form an effective NOT CONTAINS operator where * indicates wildcard characters and <i>string</i> is the text value. CONTAINS is equivalent to EQ *string* .	

WHILE...ENDWHILE structures can be nested up to ten levels deep. Use the BREAK command in an IF-block contained in a lower-level WHILE-block to pop program control back to the next outer loop when exceptional conditions are encountered in the lower-level loop.

Examples SET VAR c1 TO 0
 WHILE c1 EQ 0 THEN
 while-block command(s)
 IF c2 NE 0 THEN
 BREAK
 ENDIF
 while-block command(s)
 ENDWHILE
 next command outside the while-block

The *while-block command(s)* are executed and the *c2* condition in the IF statement is evaluated. If *c2* has a value of zero, the BREAK command is executed and the WHILE loop is terminated. The commands immediately following the ENDFILE statement are then executed.

As long as the WHILE condition (*c1*) and the IF condition (*c2*) remain true, the WHILE loop continues processing all while-block commands within the WHILE ... ENDFILE structure. When processing causes either of these conditions to be false, the WHILE loop terminates and control is passed to the commands immediately following the ENDFILE statement.

For examples of nested while-loop processing, see chapter 15 in the *R:base 5000 User's Manual*.

WRITE**SYNTAX**

```
WRITE "message" [AT scrnrow scrncol]
```

"*message*" is a string of characters to be displayed on the screen

- Purpose** The **WRITE** command, in a command file, displays a message on the screen when the command is executed.
- Options** *AT scrnrow scrncol*: This option indicates a screen row and column coordinate. It is used to locate the message on the screen at a position other than the current cursor position.
- Comments** The quotation marks which begin and end the message string are not displayed.
- The **WRITE** command displays a message. If the intent of the message is to prompt the operator for input to the command file, use the **FILLIN** command.
- You can use successive **WRITE** commands without the **AT** option in a command file and the messages will not overwrite.
- Example** **WRITE "Press any key to continue" AT 24 10 ; PAUSE**
- This command line displays *Press any key to continue* on the operator's entry screen at line 24 starting in column 10 and halts command file execution until the operator presses a key. After the **PAUSE** command suspends execution, the cursor will not necessarily be at line 25.

This Page Intentionally Blank

R:base 5000 Utilities

How to Use This Chapter	R2-2
The EXPRESS	R2-3
RBEDIT	R2-4
R_COMPILE	R2-5
The FileGateway	R2-6

HOW TO USE THIS CHAPTER

This chapter contains brief descriptions of the EXPRESS, RBEDIT, RCOMPILE, and the FileGateway. For full information see the *R:base 5000 User's Manual* chapters indicated in this chapter.

EXPRESS—The Application EXPRESS

Syntax	<p>To execute the EXPRESS program, insert the EXPRESS disk as instructed in chapter 1. At the DOS prompt, type:</p> <p>EXPRESS</p> <p>or choose from the R:base 5000 main menu.</p>
Purpose	<p>The Application EXPRESS provides a method for building R:base 5000 applications without learning how to enter R:base 5000 commands. The EXPRESS uses a visually-oriented interface to lead an application developer through the process of building a database structure, application main menu, and actions that carry out the main menu selections.</p>
Options	<p>The EXPRESS main menu offers the following options:</p> <ul style="list-style-type: none">■ Defines a new database■ Changes an existing database definition■ Builds application menus and support files■ Changes an existing application■ Displays file directory■ Exits from the EXPRESS
Comments	<p>In addition to the ready-made application actions that the EXPRESS has available, the developer can develop custom command files, menus, screens, data entry forms or reports and install them in the application.</p> <p>After the EXPRESS builds an application by collecting all parts of the application in an application file, that file can be input to the R:base 5000 compiler, RCOMPILE, for conversion into a procedure file. Applications packaged as procedure files execute faster than the application in the form of stand-alone ASCII files and is secure from user tampering.</p> <p>See chapter 14 in the <i>R:base 5000 User's Manual</i> for detailed information on the EXPRESS.</p>

RBEDIT

Syntax	To execute the R:base 5000 editor, RBEDIT, select the RBEDIT option from the R:base 5000 family menu, or type: RBEDIT at the R> prompt or the DOS prompt.
Purpose	RBEDIT, the R:base 5000 editor, is a full-screen editor for building and changing ASCII text files.
Options	When you run RBEDIT, the first menu to appear gives you the opportunity to create a new file or edit an existing file.
Comments	RBEDIT can build ASCII text files containing up to 800 80-character long lines. A line and character counter is maintained on the screen at all times as you move the cursor around the file using a set of full-screen editing keys. See chapter 15 in the <i>R:base 5000 User's Manual</i> for detailed information on RBEDIT.

RCOMPILE

Syntax	<p>To run the compiler, type:</p> <p>RCOMPILE</p> <p>in response to the DOS prompt or select the RCOMPILE option from the R:base 5000 main menu.</p>
Purpose	<p>RCOMPILE, the Compiler, provides a method of converting ASCII files that are part of an R:base 5000 application to a binary format. Command files, display text files, and menu files may be converted. After application files are processed by RCOMPILE, they run faster and are secure from user changes.</p>
Options	<p>The compiler main menu offers the following options:</p> <ul style="list-style-type: none">■ Converts a run file to binary format■ Adds a run file to a procedure file■ Adds a display file to a procedure file■ Adds a menu file to a procedure file■ Processes a file generated by the Application EXPRESS■ Displays the directory■ Displays the contents of an ASCII file■ Exits from RCOMPILE
Comments	<p>In addition to stand-alone ASCII files, RCOMPILE accepts ASCII application files (which contain command blocks, screen blocks, and menu blocks) that have been created by the EXPRESS.</p> <p>RCOMPILE collects the binary form of the ASCII command files, display files, menu files or EXPRESS application files into procedure files. RCOMPILE is also capable of transforming a stand-alone ASCII command file into a stand-alone binary file.</p> <p>See chapter 16 of the <i>R:base 5000 User's Manual</i> for detailed information on RCOMPILE.</p>

FILEGATEWAY

Syntax	<p>To execute the FileGateway program, insert the FileGateway disk as instructed in chapter 1. At the DOS prompt, type:</p> <p>GATEWAY</p> <p>or choose from the R:base 5000 main menu.</p>
Purpose	<p>The FileGateway provides a method to transfer files created by other programs and systems into an R:base database.</p>
Options	<p>Options are provided for transferring the following files:</p> <ul style="list-style-type: none">■ DIF format files from VisiCalc■ ASCII delimited files■ ASCII fixed field files■ SYLK format files from Multiplan■ dBASE II files■ Lotus 1-2-3 files■ PFS:FILE files
Comments	<p>The FileGateway is a menu-driven subsystem which prompts for all required entries.</p> <p>See chapter 13 of the <i>R:base 5000 User's Manual</i> for detailed information on the FileGateway.</p>

R:base 5000 Error Messages Contents

How to Use This Chapter	R3-2
R:base Messages	R3-3
General Processing Messages	R3-3
Column Messages	R3-4
Table Messages	R3-6
Relational Operation Messages	R3-6
SORT Clause Messages	R3-7
WHERE Clause Messages	R3-7
General Variable Messages	R3-8
Defining Variables	R3-8
Using Variables	R3-9
DOS and System Messages	R3-10
Command Specific Errors	R3-12
R_COMPILE Error Messages	R3-35
FileGateway Error Messages	R3-36
EXPRESS Error Messages	R3-42

HOW TO USE THIS CHAPTER

This chapter of the *R:base 5000 Reference Manual* contains lists of error messages produced by R:base, RCOMPILE, the Application EXPRESS, and the FileGateway.

R:base error messages are divided into groups which reflect the operational areas of R:base. These groups are:

- **General Processing Errors:** Within this group, errors are divided into categories. These categories include general R:base messages, general column messages, general table messages, relational operation messages, SORT clause messages, WHERE clause messages, general variable messages, DOS and other system messages, and database status messages.
- **Command Specific Errors:** Defines errors which are produced only when executing a specific command. The commands which have specific messages are listed alphabetically with the errors produced by each listed alphabetically under each command. If you receive an error message while executing commands at the R> prompt or any of the R:base modes (DEFINE, HELP, PROMPT, REPORTS, FORMS, LOAD), then look here for information.

Some error messages have four digits beginning with 2 (2534, for example). These messages are combination messages--that is, two or more phrases are combined to generate the message. Most of these combined messages generate a -1 when trapped in an error variable. See SET ERROR VARIABLE in the "Command Dictionary" section of this Reference. A few of these messages can be trapped when generated from a command file.

For RCOMPILE, the EXPRESS and the FileGateway, messages are grouped alphabetically.

In R:base, messages may occur when a command is executed either at any of the R:base prompts or when a command is executed from a command file. Non-fatal messages can be suppressed by executing the command:

SET MESSAGES OFF

Many error messages which occur when a command is executed from a command file can be trapped using the SET ERROR VARIABLE command like this:

SET ERROR VAR varname

Errors cannot be trapped in the DEFINE, LOAD, HELP, and PROMPT modes. Each error has a unique message number that is listed in this chapter in parentheses next to the error text. For example,

Illegal database name (9)

If you had set an error variable using the SET ERROR VARIABLE command, then your command file code could test for the error variable value. If the value was 9, then you would know that the database name was not a legal name. You could write command file code to recognize this particular error and execute alternative code if the error occurs. You can anticipate these kinds of errors and add error-handling code to the command file to avoid unintentional exits from the application.

Some error messages have more than one message number, for example:

-WARNING- No rows satisfy the WHERE clause (56/137)

This only means that the error message might have either number if trapped in an error variable. If you are trapping errors in a command file, check the error variable for both numbers.

R:BASE MESSAGES

General Processing Messages

(no message) (0)

Error variables have a value of zero if no error condition is present.

Illegal database name (9)

Database names must be 1-7 characters and may include a drive designation and path, for example: *d:\mydb\dbname*.

Illegal file name (32)

File names must be 1-8 characters and may include a drive designation and path, for example: *d:filename.ext*.

Incorrectly specified (ROW,COL) (301)

You used an AT clause to specify a screen row and column. Check that the screen row is between 1 and 24 and that the screen column is between 1 and 79.

Input file not found (33)

R:base could not locate the specified input file. Check to see that you spelled the file name correctly and have included a drive designation and path if the file is not located on the current directory.

Missing database name (8)

The command you entered required a database name or you did not indicate a drive and directory if the database is not on the default drive and current directory.

More than 1600 characters on input line (14)

You have attempted to enter more than 1600 characters as input to R:base. Command lines are limited to 1600 characters including all spaces and continuation characters (+).

Number is out of range (565)

The screen column or row number specified with an AT clause is out of range. Screen rows may be 2-24 and screen columns may be 1-79.

Reserved words cannot be used for names (57)

You are attempting to use an R:base command or other reserved word as a table, column, or variable name. Choose another name.

Switching INPUT back to KEYBOARD (26)

The routine currently reading from an external file failed. R:base automatically switches input back to the keyboard. Check your command or procedure file for errors.

Syntax is incorrect for the command (39)

The command was not properly entered. Check the displayed syntax and reenter the command properly. Make sure that the command keyword/argument combinations were entered in the proper order and that all required parts of the command were entered.

Unrecognized command - retype it (40)

You did not enter a valid R:base command. Check to be sure that you correctly spelled or abbreviated the command and that the syntax is correct.

COLUMN MESSAGES

Allowable column types are TEXT, INTEGER, REAL, DATE, TIME, and DOLLAR (194)

The second item of a column definition, data type, must be one of the reserved words shown in the message.

-ERROR- Column <colname> is not in the table <tblname> (2045)

Verify that you spelled the column and table names correctly. Use the LIST command to check for valid table and column names.

-ERROR- Column name <colname> is used more than once (2053)

The specified column name is used more than once. This usually occurs in a USING clause or when defining a table.

-ERROR- Column <colname> is not in the table (2045)

The column you specified does not exist in the specified table in the database. Verify to be sure you spelled the column name correctly.

-ERROR- Column <colname> is not defined (2534)

You have attempted to use a column name which does not exist in either the specified table or, for some commands, any table in the currently open database. Verify that you spelled the column name correctly and have specified the correct table.

Illegal column name (47)

The column name you have specified is not valid. Column names must be text items one to eight characters long.

New table has too many columns (52)

You have created a new table which has exceeded the 400 column limit on the database.

Too many columns in the database - limit is 400 (50)

The full capacity of R:base is 400 columns in one database. You must remove one or more columns from the database before you can define a new column.

-WARNING- Length after INTEGER or REAL defines an ARRAY. Please enter <colname> without length if you do not want an array (2058)

You have entered a length designation with a column or variable definition when the column or variable is an INTEGER or REAL data type. If you did not intend to define an array, reenter the definition without the length.

TABLE MESSAGES

Illegal table name (42)

An item in a command line which is supposed to be a table name must be a text item one to eight characters long.

New table name is a duplicate (55)

You have attempted to define a table which already exists in the database. Use a new table name.

New table is wider than 1530 bytes (49)

The maximum length of a row is 1530 characters and the table you are attempting to add will make the row longer than 1530 characters. Remove any unneeded columns from the table.

<tblname> is an undefined table (43)

The named table does not exist in the currently open database. Check to be sure you entered the correct table name.

Table not defined (90)

The table you are attempting to use is not defined in the currently open database.

Too many tables defined - limit is 40 (48)

You may define no more than 40 tables for a database. Use the REMOVE command to delete unneeded tables.

-WARNING- No data exists for this table (2136)

You are attempting to process data in a table which contains no rows.

-WARNING- Unauthorized access to the table (41)

The table is password protected and you have not entered the proper password to read or modify the table. Enter your password with the USER command.

RELATIONAL OPERATION MESSAGES

Column <colname> is not common to either table (2277)

The column you specified in a USING clause does not exist in either specified table. Check to be sure you spelled the column name correctly.

-ERROR- Tables <tblname1> and <tblname2> have no common column names (2279)

The tables you specified do not have a common column name. A common column name is required to complete the operation.

SORT CLAUSE MESSAGES

More than 10 sorted columns (51)

You have specified more than 10 columns in a SORTED BY clause. The limit is 10.

WHERE CLAUSE MESSAGES

All values in qualifying rows are missing (139)

The rows defined by the WHERE clause do not contain any data. Be sure that your WHERE clause conditions are correct.

Cannot recognize WHERE operator (622)

R:base does not recognize the logical operator you are using in the WHERE clause. R:base recognizes the following conditional operators: EXISTS, FAILS, EQ, NE, GT, GE, LT, LE, CONTAINS, EQA, NEA, GTA, GEA, LTA and LEA.

<colname1> and <colname2> data types cannot be compared (2076)

You have attempted to compare two columns which do not have the same data type.

-ERROR- Column <colname> must be compared to <datatype> value (2071)

You have attempted to compare a column to a value with a different data type. The comparison value must have the same data type as the column to which it is compared.

Illegal boolean operator between column name and value (70)

R:base does not recognize the operator you are using in the WHERE clause. R:base recognizes the following conditional operators: EXISTS, FAILS, EQ, NE, GT, GE, LT, LE, CONTAINS, EQA, NEA, GTA, GEA, LTA and LEA.

Illegal limit specification (75)

You have attempted to use a WHERE clause in the form *WHERE LIMIT = n* and *n* is not an integer.

Too many conditions in the WHERE clause (624)

You have attempted to use more than 10 conditions in a WHERE clause.

-WARNING- No rows satisfy the WHERE clause (56/137)

Your WHERE clause conditions cannot be satisfied. Therefore no rows have been selected. Check that your WHERE clause conditions are valid.

WHERE clause is incomplete (623)

You have entered an incomplete WHERE clause. If you specify a condition, it must be complete—that is, a comparison between columns, a column and a value, a column and a dotted variable, or one of the special conditions such as LIMIT.

GENERAL VARIABLE MESSAGES

Defining Variables

Cannot recognize second operand (496)

You have attempted to define a variable. R:base does not recognize the second operand in the variable expression. Verify that you used the correct syntax and spacing in the command.

Expression cannot be evaluated (453)

R:base cannot evaluate the variable expression. Expressions must be in the form *varname* = *operand op operand* or *varname* = *value*. Verify that you used the correct syntax and spacing in the command. See SET VARIABLE in the “Command Dictionary” chapter of this Reference for information.

Invalid operator (495)

You have attempted to use an invalid operator in a variable expression. See SET VARIABLE in the “Command Dictionary” for information.

Too many variables defined. Press any key to continue (601)

You have too many global variables defined. Use the CLEAR VARIABLE command to remove unneeded global variables.

Variable table is full, use an existing variable or CLEAR a variable (490)

Variable limit is exceeded (509)

You have attempted to define a variable and you already have defined too many global variables in your database. Use SHOW VARIABLES to display your variable table and CLEAR VARIABLES to remove unneeded variables.

Using Variables

Error variable not found (637)

R:base could not find the specified error variable. See SET ERROR VARIABLE in the "Command Dictionary" for information.

The Error Variable must be an INTEGER (85)

You have attempted to use a non-integer variable as an error variable. You must first define the variable as INTEGER with SET ERROR VARIABLE to use the variable to capture error message numbers.

Type conversion error changing <varname> to type <datatype> (2529)

You have attempted to change the data type of an existing global variable but the variable value does not match the defined data type.

Variable <varname> is not defined (2020)

You have attempted to use a global variable which is not defined. Use SHOW VARIABLE to list existing global variables. Use SET VARIABLE to define new global variables.

Variable must be an integer (492)

You have attempted to use a variable which is not data type INTEGER to COMPUTE ROWS or COUNT columns.

Variable not defined (488)

You have attempted to use a variable which is not defined. Use SHOW VARIABLE to list existing global variables. Use SET VARIABLE to define new global variables.

Variable type does not match column type (484/635)

You have attempted to compare a variable to a column, or you have attempted to enter column data into a variable with a different data type than the column. The column and variable data types must match either to compare them or to place the value of the column into the variable.

Your value does not have the same type as your variable (480)

You have attempted to compare a variable to a value, or you have attempted to enter a value into a variable with a different data type than the value. The value and variable data types must match either to compare them or to place the value into the variable.

DOS AND SYSTEM MESSAGES

Disk full (571)

R:base cannot write information to a full disk.

Disk I/O error in PACK command (220)

When you used the PACK command, R:base was unable to either write to or read the disk. Use the DOS CHKDSK command to validate the disk.

Disk problems--Please check disk and files. (44)

Verify the quantity of free space on the disk. If your disk is full, delete unneeded files or use the PACK command to release unused space.

Error in closing (19)

R:base could not properly close a file.

Error in writing (15)

R:base could not properly write a file.

File not found (526)

R:base could not find the specified file. Verify that you spelled the file name correctly and that you included a drive and directory designation if the file does not reside on the default drive and directory.

Internal error in the pointers in File 2 (67)

Your database has possibly experienced severe damage. Refer to your back-up copy.

Internal error referencing variable (609)

An internal R:base variable reference has been damaged. Refer to your back-up copy.

Invalid directory (527)

You have entered a non-existent directory or path. Be sure you are attempting to access an existing directory and have spelled each directory in the path correctly.

Invalid drive (528/576)

You have attempted to access a non-existent drive. On a two floppy disk system, the drives are *a:* and *b:*. On most hard disk systems, the floppy drive is *a:* and the hard disk is *c:* with any additional floppy or hard drives having other one character drive designations.

I/O error on OUTPUT file - check for full disk (65)

You have attempted to write a file to disk and R:base has found no room on the designated disk for the output data. Use another floppy disk or delete unneeded files from the disk. If the message was presented while writing to hard disk, you must delete files to make space. You can also use the PACK command on your databases to free unused disk space.

Out of file handles (551)

The command FILES=20 is not in your *CONFIG.SYS* file. See chapter 1 for installation instructions.

Sorting file problems - Check for full disk (500)

When R:base sorts a table, it creates a temporary sort file on the default drive and current directory. Your default disk does not have enough room to create the temporary sort file. Delete unnecessary files, and use PACK on your databases to release unused space.

Unable to close database (10)

R:base could not successfully close an open database. This message usually indicates a disk problem—possibly a full disk. Use CHKDSK to validate the disk drive and to check for free space. Use PACK on your databases (after making a back-up copy) to release unused space. Delete unneeded files.

Unable to open database (7)

R:base could not open the specified database. Verify that you spelled the database name correctly.

Unable to open file (31)

R:base could not open the specified file. Verify that you spelled the database name correctly and, if the database is not on the current directory, specified a drive and directory name.

Command Specific Errors

ASSIGN

ASSIGN command requires INTEGER, REAL, or DOLLAR column (250)

You can change the value of an INTEGER, REAL, or DOLLAR column only.

Cannot assign array column (251)

The ASSIGN command cannot be used to assign values to arrays because arrays are not used in R:base.

Expression cannot be evaluated (252)

You have used an expression that is not valid. Check both your expression operator and column type, and see if the combination makes sense.

Operator must be + - X / or % (253)

The operators +, -, X, /, and % are the only valid operators to use in the ASSIGN expression.

CHANGE

A WHERE clause is required (247/399)

To protect you from inadvertently making a global change, R:base requires you to specify a WHERE clause for the CHANGE command. For a global change, specify a WHERE clause that qualifies all rows. For example:

WHERE colname EXISTS

CHECK/NOCHECK

You must use the OWNER password to change RULES checking (116/214)

You have attempted to change checking by using the SET RULES or the CHECK/NOCHECK commands. The USER and OWNER passwords must match to change RULES checking.

CHOOSE

Invalid data in the menu file (593)

The menu file contains invalid data. See chapters 14 and 15 for information on menu files and valid data formats for menu files.

Not a menu procedure (592)

The menu name you specified does not refer to a menu procedure in the procedure file you specified.

COMPUTE

Function must be COUNT, MIN, MAX, SUM, AVE, ROWS, or ALL (149)

R:base does not perform the function you attempted. Functions are: COUNT, MIN, MAX, SUM, AVERAGE, ROWS and ALL.

SUM and AVE only work with INTEGER, REAL, and DOLLAR columns (150)

You are trying to compute the sum or average of a column which has a data type that is not INTEGER, REAL, or DOLLAR.

DEFINE

Columns must be defined before tables (181)

In the DEFINE mode, you must first define columns and then tables. Tables cannot contain undefined columns.

Valid DEFINE submodule commands are:

OWNER, COLUMNS, TABLES, PASSWORDS, LIST, NEWPAGE, PROMPT, RULES, or END

You have entered a command in DEFINE mode which R:base does not understand. The message lists the valid submodules available in DEFINE mode.

Columns

Allowable column types are TEXT, INTEGER, REAL, DATE, TIME, and DOLLAR (2194/2536)

The second item of a column definition, data type, must be one of the data types shown in the message.

Column length must be between 1 and 1500 (196)

The maximum length specification for a TEXT column in a column definition is 1500 characters. The minimum specification is one character.

Existing column names cannot be redefined (193)

A column that has already been defined and has been included in a table cannot be redefined by using the COLUMN command.

Illegal column name (47)

The first item of the column definition, *colname*, is not a character string of one to eight characters in length.

Reserved words cannot be used for column names (57)

R:base reserves certain character-string values for system use.

The above column names will be lost if you leave the DEFINE module (2201)

The columns listed are not included in a table. If you do not include those columns in a table, the column definitions are not be saved.

The following columns are not associated with a table (2199)

The columns listed are not included in a table.

You cannot specify a length for DATE, TIME, or DOLLAR (197)

There are too many items in the column description. You cannot specify a length for a DATE, TIME, or DOLLAR data type.

Tables

Column <colname> is already used in this table (2189)

You have already included this column in the table you defined.

Columns must be defined before tables (181)

Column names, data types, and lengths must be defined using the COLUMNS command in DEFINE mode before a table may be defined.

<colname> is an undefined column name (94)

R:base could not locate the specified column among the currently defined columns of your existing tables. Tables must be built with defined columns. Columns are defined using the COLUMNS command in DEFINE mode.

-ERROR- Column <colname> not defined (2534)

You are attempting to add an undefined column to a table. A column must be defined before you can include it in a table.

Illegal table name (42)

The first item of the command line when defining a table must be a table name that is 1 to 8 characters long.

New table is a duplicate (55)

You are attempting to define a table that already exists in the currently open database.

Too many tables defined - limit is 40 (48)

You cannot define more than 40 tables.

New table is wider than 1530 bytes (49)

The table you are trying to define would result in more than 1530 bytes per row with the combination of columns specified. Make sure that you indicated only the columns you wanted and that they have proper column lengths defined.

Owner

Illegal OWNER password (185)

The second item in the OWNER command is not a text item that is 1 to 8 characters in length

-WARNING- An OWNER password should be defined when using passwords (62)

You entered the PASSWORD command in the DEFINE mode. Unless you first define an OWNER password, the passwords you define do not protect your data.

-WARNING- This password will not allow you to access any table (63)

The password you entered with the USER command is not valid.

You must enter the correct OWNER password (186)

You entered an incorrect password. The database you are defining has an OWNER password that does not match your current USER password.

Passwords

Illegal password name (198)

The fifth item of the command line, the password, is not a text item between 1 and 8 characters.

Rules

Columns must be the same type and length (95)

Compared columns must have the same type and length.

Incompatible RULES table already exists (102)

The table named rules in your database does not contain R:base rules information.
Rename the table before attempting to define data entry rules.

Insufficient space to process rules (101)

There is not enough RAM for the R:base rule checking function to verify data input.
You have defined too many loops in the WHILE structure or too many global variables.

No database open (93)

You are attempting to define rules without an open database.

Rule components must apply to the same table (91)

All columns which are checked in a single rule must be in the same table.

Rules must be joined with AND or OR (97)

To combine rules on a single command line, you must use either AND or OR to join component rule parts.

Too many columns to define rules (104)

You are limited to 400 column names throughout all tables, forms, reports, and rules.

Too many tables to define RULES (103)

Rules are stored in a table. You cannot define rules because you already have 40 tables in the database.

Table must be specified (98)

Rule entries that use the IN option require that a table name follow the keyword IN.

Table not defined (90)

When a rule refers to a table, that table must exist in the database.

Too many conditions for a rule. The maximum is 10. (2099)

You have attempted to define too many rule conditions. The maximum is 10.

Unrecognized Boolean expression (92)

You attempted to join two rule conditions with a word other than AND or OR.

Value incompatible with column (96)

The rule specified does not match the data type previously defined for this column.

-WARNING- Value truncated to 40 characters (89)

The error message you define for a rule must be less than 40 characters. R:base truncates messages longer than 40 characters.

DELETE

A WHERE clause is required (247/399)

This message tells you to add a WHERE clause to your DELETE ROWS command. The purpose of this WHERE clause is to prevent inadvertent global changes.

DISPLAY

Not a display procedure (568)

The screen name you specified is not a display procedure in the procedure file you specified.

DRAW

Cannot DRAW any part of a FORM above or below screen limit (604)

R:base cannot draw a variable FORM above line 2 or below line 24 on the screen.

EDIT

-WARNING- No data exists for this table (428)

The table you have specified does not have any data loaded.

EDIT USING

< formname > is an undefined screen FORM (401/427)

You are trying to edit data using a FORM that does not exist.

Insufficient space to perform EDIT (429)

The table has more rows than R:base can fit into the temporary edit memory. Split the table into two or more smaller tables to edit. Then recombine the tables after editing.

No FORMS have been defined (426/605)

No data input FORMS have been defined for editing data.

< tblname > is not a valid table (612)

The table specified is not a valid table name. Verify that you entered a valid 1-8 character table name.

-WARNING- No data exists for this table (428)

The table you specified does not have any rows of data.

ENTER

< formname > is an undefined screen FORM (401/427/607)

You specified a non-existent FORM. Verify that you spelled the FORM name correctly and that the FORM exists in the currently open database.

Illegal file name (403)

The file you have specified is not a valid DOS file name.

Variable <varname> not defined in the FORM (2610)

When you enter the ENTER VARIABLE, EDIT VARIABLE, or DRAW commands, the variable list includes a variable that is not located on the FORM being used.

EXIT

Unable to close database (10)

R:base was not able to write data to the database files when closing the database.

EXPAND

Allowable column types are TEXT, INTEGER, REAL, DATE, TIME and DOLLAR (2536)

You have attempted to add a column with the EXPAND command and have not entered a valid data type. The valid types are listed in the message.

Cannot execute <command> with more than 39 tables defined (2531)

The command you are attempting to execute creates a temporary table and you already have 40 tables defined. Remove unneeded tables from your database.

Column length must be between 1 and 1500 (538)

The maximum length of a TEXT column is 1500 characters.

Column makes table wider than 1530 bytes (540)

The maximum length of a row is 1530 characters and the table you are attempting to add will make the row longer than 1530 characters. Remove any unneeded columns from the table.

-ERROR- Column <colname> is already used with this table (2541)

You are attempting to add a column to a table and the table already contains a column with the same name. Choose another name for the column if necessary.

Too many column names in the database – limit is 400 (548)

You are attempting to add a column to a table and you already have 400 columns defined. Remove any unneeded tables and columns from your database.

You cannot specify a length for DATE, TIME or DOLLAR (539)

There are too many items in the column description. If the second item in a column description (*datatype*) is DATE, TIME, or DOLLAR, then there does not need to be a third item (*length*) in the description because R:base assigns a default length.

FORMS

Column < colname > must be a valid < datatype > (2436)

You are attempting to enter an invalid data type into a column. For example, a column defined as a DATE data type cannot accept TEXT data (which is not a valid DATE format).

FORM definition is incomplete (440)

You are trying to edit a FORM that is incomplete because it is missing columns specification.

Incompatible FORMS table already exists (418)

A table named FORMS already exists in the database, however, it does not have the FORMS structure. If the database was originally developed in R:base 4000, check that you use the CONVERT program (see chapter 1). If you created a table named FORMS for some other purpose, rename it to something else.

Layout data is incorrect – check it out (441)

Some part of the FORM layout is not valid. This may mean that the FORM uses columns that have been removed from the table.

Too many columns to define FORMS (420)

FORMS uses two column names, FNAME and FDATA. If your database already has 400 columns, you have cannot define FORMS until you remove at two columns. Delete unused column names or tables.

Too many tables to define FORMS (419)

FORMS is an R:base table. If 40 tables exist in a database, you cannot define FORMS unless you remove a table.

Variable is not found (563)

A variable specified in DRAW, EDIT VARIABLE, or ENTER VARIABLE is not on the FORM.

You do not have any column names defined in this FORM (444)

You must locate at least one column on the FORM.

You don't have modify permission on FORMS (494)

Your modify password (MPW) does not match the designated modify password for the open database. Enter the correct password to change the FORM.

You have duplicate layout entries (443)

You have the same column or variable located twice on the same FORM.

You have more than 23 lines on the FORM (439)

You have used more than 23 lines on the FORM layout. The line limit is 23. Adjust your FORM layout so that all columns can be represented on 23 lines or define more than one variable FORM and use the DRAW command to allow multi-form entry.

You have too many column name layouts (442/448)

You have specified locations for more columns than previously defined for the table or you have more than 23 lines on the input FORM. You are allowed 23 lines to define a FORM for data input.

GOTO

Unable to find matching LABEL for <labelname> (594)

The labelname specified by the GOTO does not exist in the command file. Check that you spelled the label name correctly and have a LABEL command with the indicated label name.

HELP

There is no help text for the <commandname> (30)

The help text file contains no entry for the command specified.

The HELP text file is not available (29)

The HELP.RBS file is not on the default drive or defined PATH. See chapter 1 for installation procedures.

IF...THEN...ELSE...ENDIF

No matching ENDIF or ENDWHILE found--Check command file code (464)

Your command file attempted to exit an IF...ENDIF structure by executing a GOTO command that referred to a LABEL outside of the IF...ENDIF structure. Restructure your command file logic so that all IF...ENDIF structures terminate only by execution of the ENDIF command.

ENDWHILE or ENDIF missing in an input file (462)

You have an IF...THEN command line without a matching ENDIF line. Check your command file. See the "Command Dictionary" for information.

Illegal Boolean operator between variable and value (22)

An operator used in the conditions that make up a boolean expression was not recognized by R:base. R:base recognizes the following boolean operators: EXISTS, FAILS, EQ, NE, GT, GE, LT, LE, CONTAINS, EQA, NEA, GTA, GEA, LTA, LEA, =, >, <, <=, >=, <>, =A, >A, <A, <=A, >=A, and <>A.

No WHILE or IF blocks are open on current input source (466)

R:base has found an ENDIF statement with no matching IF.

WHILE and IF commands are nested beyond maximum (465)

You may have a maximum of 10 nested IF...ENDIF structures in a command file and you have exceeded this number. Restructure your command file.

INPUT

Illegal file name (32)

The name you specified is not a valid DOS file name.

Input file not found (33)

The file you specified does not exist. Be sure you spelled the file name correctly and included drive and path specifications if the file is not on the current directory.

Unable to open file (31)

The file you specified does not exist. Be sure you included the drive and path specifications if the file is not on the current directory.

JOIN

Joining columns differ in type or length (310)

The two columns you specify in the USING clauses must be the same type and the same length.

-WARNING- < COLNAME > is a duplicate column name. You should rename one of them before using (2307)

Two columns in the new table have the same name. Change the name of one of the columns so you can refer to each with a unique name within the new table.

LIST

-WARNING- No columns are defined (177)

-WARNING- No tables are defined (166)

R:base cannot find any columns or tables to list. You must open a database before using the LIST command.

LOAD

Cannot LOAD with PROMPTS and an AS clause (127)

You are not allowed to load data from the keyboard with prompts and use an AS clause.

Cannot LOAD WITH PROMPTS and FROM a file (128)

You are not allowed to load data from a file and load with prompts at the same time.

Incorrect number of values for this table (118)

Unless you have specified FILL, you must enter a value for every column in the table. The message indicates you have not entered enough values.

Incorrectly specified (ROW,COL) (131)

You have included an AS MPLAN clause in the LOAD command but have given an incorrect row and column specification.

Missing table name (126)

There must be at least two items on the command line, the second being the name of the table.

No data exists at the specified row and column (455)

You are attempting to load a Multiplan SYLK file and the offset (row, column) specified is a place in the worksheet where no data exists.

No FILE specification (129)

You omitted the file specification from the LOAD command. R:base gives this message when an AS clause is included but no file specification is given.

The USER and OWNER passwords must match (133)

To load data into a table, you must have matching USER and OWNER passwords.

Unidentified AS clause--Use the FileGateway for SYLK and DIF files (132)

You have included an AS clause in the LOAD command but R:base does not recognize the item following AS. The AS clause may be: AS ASCII, AS DIF, or AS MPLAN.

-WARNING- Column <colname> will be truncated (2123)

You have entered a string with more characters than the length limit defined for the columns. R:base fills as many characters as you have previously defined.

-WARNING- Excess data values ignored (125)

You entered more data values than are needed for one row of the table. All excess data values are ignored.

-WARNING- Value truncated to 40 characters (89)

Part of the data in the input file was outside the length specification of the column. The value was truncated at 40 characters.

MOVE

Number of characters must be 1 to 1500 (491)

The parameters *nchar*, *chrpos1*, and *chrpos2* must all be integers between 1 and 1500.

OPEN

Missing database name (8)

You did not enter a database name with the OPEN command. Refer to the R:base prompt menu or help text for the proper syntax.

Illegal database name (9)

The second item of the OPEN command must be the name of an existing database. Verify the spelling of the name.

Unable to open database (7)

R:base is unable to open the files which correspond to the name of the database. Make sure you properly specified the name and that the corresponding files do exist.

OUTPUT

Can't create destination file (570)

R:base cannot create the specified destination file name. Check that you specified a valid drive and directory for the file.

PACK

Disk I/O error in pack command (220)

While attempting to execute the PACK command, R:base experienced a drive or disk problem. Use the CHKDSK command to check the validity of the database drive and disk.

PRINT

Errors in variables. Correct with REPORT command (13)

R:base found error conditions in the defined report variables. Use REPORTS to check the validity of defined variables.

< rptname > is an undefined REPORT. (138)

The specified REPORT on the command line has not been defined for the open database. Check the LIST REPORTS output.

PROMPT

There is no PROMPT text for the command < command > (83)

The command you needed help with is one that is not available from PROMPTS. Refer to the help file or the *R:base 5000 User's Manual* for proper use of the command.

The PROMPT file is not available (82)

To be available to the cursor, the PROMPT.RBS file must be located on the default drive or found in a subdirectory indicated by the DOS PATH command. See chapter 1 in the *R:base 5000 User's Manual* for installation procedures.

RBEDIT

File too large for RBEDIT. (512)

The stand-alone version of RBEDIT can edit a file of up to 800 lines when it has the maximum amount of memory available. When executed from within R:base, the maximum number of lines is approximately 130.

RBEDIT cannot make this file any bigger. (219)

RBEDIT has run out of memory. Save the current file. Try to cut down the file size by subdividing the file into several smaller files.

RELOAD

Illegal database name (318)

Your database name is longer than seven characters. The database name specified must be a text of less than eight characters.

Missing database name (317)

You must specify a new database name to perform a reload of your current database.

Unable to open reload file (316)

R:base was unable to open a scratch file because there was not enough room on the disk.

Write error on reload file – Check for a full disk (319)

During the reload operation, R:base was not able to write out all reloaded blocks of data. Make sure your disk has room for the new data.

REMOVE COLUMN

Cannot use the REMOVE command with only one column in the table (2546)

You cannot remove the last column in a table. Use the REMOVE table form of the command instead to delete the table.

RENAME

< colname > is in another table. Use a different name (545)

You have attempted to rename a column to a name which already exists in another table. Choose a new column name.

New name is already being used for a column which differs in type or length (2273)

You have attempted to rename a column to a name which already exists with a different type and length. Choose a new column name.

New name is a duplicate (275)

You have attempted to rename a column and the column name already exists in the same table. Choose a new column name.

Owner password is incorrect (267)

You have attempted to RENAME a table or column and your USER password does not match the OWNER password for this database. The two passwords must match to RENAME.

RENAME file

Unable to rename file (550)

R:base could not rename the specified file. Verify that you spelled the file name correctly and have specified the drive and directory if the file is not on the current drive and directory.

REPORTS

Cannot recognize equal sign (=) (387)

REPORTS cannot find the equal sign in the variable expression. Proper syntax is:
varname = expression.

Cannot recognize first operand (388)

REPORTS cannot find the first operand in the variable expression. Proper syntax is:
varname = operand1 op operand2.

Cannot recognize number (368)

REPORTS does not recognize the position specified in a REORDER command. The position must already exist as a variable definition.

Cannot recognize operator (390)

REPORTS does not recognize the operator entered in the variable expression. Valid operators are: +, -, x, /, %, and SUM OF.

Cannot recognize type (366)

REPORTS does not recognize the data type you have specified for a variable when using TYPE in the *Define* option. Valid data types are DATE, DOLLAR, INTEGER, REAL, TEXT, and TIME.

Cannot recognize variable name (350/381)

REPORTS cannot interpret the variable name. Variable names must be 1 to 8 characters.

< colname > is not a column in table < tblname > (613)

You have attempted to define a lookup variable with a reference to a non-existent column.

-ERROR- Already 40 variables defined (380)

You are attempting to add a variable and the maximum number of variables have already been defined. Use DELETE to remove unneeded variable expressions.

-ERROR- No column names or variables have been located (349)

You have requested the *Relocate* option and no columns or variables have been located. Use the *Locate* option instead.

-ERROR- Maximum number of column names, variables located (505)

You are attempting to locate another column or variable and the maximum columns (number in the table) and 40 variables have already been located on the REPORT format.

-ERROR- Not a column or variable name (355)

You have attempted to a variable or column which does not exist. Press [F3] to display a list of valid columns and variables.

-ERROR- Pagesize must be less than 1000. (510)

You have attempted to set a page length longer than 999. If you do not want page breaks, enter zero (0) as the page length.

-ERROR- There are no variables defined. (363)

You have attempted to DELETE a variable and no variables are defined.

Error(s) in variables. Correct with Define. (356/498)

This error occurs when you are entering REPORTS mode with an existing report. Something has corrupted your variable definitions (for example, you have removed a column which was used in a variable expression). Request the REPORTS *Define* mode to correct variable errors.

Illegal character in variable name (383)

The variable name you are attempting to use contains illegal characters. Variable names must be alphanumeric only.

Incompatible REPORTS table already exists (344)

You are trying to define a REPORT name that already exists as a table. Enter a new name.

Insufficient space to process REPORTS command (504)

There are too many global variables defined, or if operating within a command file, 10 WHILE...ENDWHILE structures are defined. If either, or both, of these cases are true, there is not enough memory left to define reports. Finish your current command file processing and then use the CLEAR VARIABLES command before attempting to define or edit reports.

Internal error – Heap allocation error (590)

There is insufficient memory available to define any more variables. You must first CLEAR any unneeded global variables before you can define more variables in the report.

Internal error – Out of dynamic space (595/589/605)

You have exceeded your available memory space. You must exit from reports and CLEAR any unneeded global variables.

Invalid result type for expression (367)

You have attempted to change the data type of a variable using the TYPE command with the *Define* option to an invalid data type for the variable.

No database open (325)

You are attempting to define a report when no database is open. Be sure to open the correct database.

Number greater than the number of variables (370)

You have attempted to REORDER variables and have specified a position which is greater than the number of defined variables.

Number less than <n> (371)

You have attempted to REORDER a variable and the position you specified is not a valid integer between 1 and the number (*n*) of defined variables.

<operator> is not a valid operator (614)

You have attempted to define a lookup variable with a WHERE clause containing an invalid operator.

REPORT definition is incomplete (446)

When editing the REPORTS table, you receive this message if you are missing some column specifications in the layout of the REPORT definition.

REPORT data is incorrect – check it out (447)

The layout defined for your REPORT is incorrect. Check that you have not changed any column names or the table for the REPORT since first defining the report.

Some expressions cannot be evaluated (336)

REPORTS cannot process all of the variable expressions. This is sometimes caused when a column name used in the expression is changed or if the a variable is used in the expression but is not yet defined. Confirm the column names for validity and make sure you have the variable expressions in the correct order.

Too many columns to define REPORTS (346)

You are exceeding the limit of 400 columns by defining the REPORTS table.

<tblname> is not a valid table (612)

You have attempted to define a lookup variable with a reference to a non-existent table.

Too many tables to define REPORTS (345)

Because REPORTS is a table, it is included in the 40 table limit for a database.

Variable cannot have the same name as a column (385)

You have attempted to define a variable with the same name as a column in the specified REPORTS table. Choose a different variable name.

Variable exists. Redefine it (Y/N)? (382)

You are redefining a REPORT variable. Press [Y] to redefine. Press [N] if you did not want to redefine.

Variable name cannot be a keyword (384)

You have attempted to define a variable with an R:base reserved word as a name. Reserved words include all command names, data types, and required table names such as REPORTS.

Variable not defined (369)

You have attempted to REORDER, TYPE, or DELETE a variable definition which does not exist.

Variables must be 8 characters or less (326)

Variable names must be from 1 to 8 characters long.

You don't have permission to modify REPORTS (474)

The USER and OWNER passwords must match before you can modify an existing REPORT format.

RUN

Can't open source file (569)

R:base could not open the specified command file. Check that you spelled the file name correctly and that your file specification is complete. Your file specification must include a drive name if the file is not on the default drive and current directory.

Display or run procedure not found (566)

R:base could not find the specified procedure within the specified run file. Check that you spelled the procedure name and file name correctly. Check that you included the drive and/or directory with the command file name if it is not on the current drive and directory.

Input file not found (33)

The file you specified does not exist. Be sure you spelled the file name correctly and included drive and path specifications if the file is not on the current directory.

Not a run procedure (567)

The procedure you have specified is not a run procedure (it may be a menu or screen). Check that you spelled the procedure name correctly and the procedure is to be RUN instead of DISPLAYed.

Unable to open file (31)

The file you specified does not exist. Be sure you include the drive and directory if the file is not on the current directory.

Variable limit exceeded (509)

You are attempting to use a RUN command with more than nine substitution parameters.

SET

Illegal date format specification (206)

The date format you want to enter is not valid. Check DATE for the format rules.

No color by that name (485)

You have attempted to set a foreground or background with a color that R:base does not recognize.

Screen width must be greater than 40 and less than 256 (207)

The screen width must fall anywhere between the indicated values.

That color can't be used as background (487)

The specified COLOR is not available as a background color.

Variable limit has been reached (209)

You are attempting to define a global variable using SET VARIABLE and 40 variables already exist. Use CLEAR to delete unneeded global variables.

Variable must be an integer (564)

An error variable must be INTEGER.

-WARNING- Unrecognized SET keyword (12/205)

Check the proper syntax for the SET command. Please refer to the Prompt menu or the Help text for the proper use of this command.

You can't have the same foreground and background (486)

You have attempted to set both the foreground and background colors the same. You should choose different colors for foreground and background.

You must use the OWNER password to change RULES checking (116/214)

Your user password must match the OWNER password to turn off rules checking.

SET POINTER

Invalid route number or route does not exist (493)

You have specified a route number that is not 1, 2, or 3. Or, if the route number is correct, there are no rows which qualify for the route.

SHOW

Invalid show parameter (639)

You have attempted to use SHOW with some other parameter than a valid *keyword* or VARIABLES.

-WARNING- No variables defined (213)

You have attempted to show the defined global variables and R:base is reporting that there are no variables.

TALLY

Too many values for TALLY (154)

You have more data than can be tallied in one pass with the current buffer space. Try doing your tally in phases by using a WHERE clause which can restrict the values of the tallied column as it is sorted. For example, if you are tallying *name*, first use a WHERE clause which asks for *name LE fzzzz*. Next, use a WHERE clause which asks for *name GE gaaaa* and *name LE mzzzz*, and so on until you have gone through all values for *name*.

UNLOAD

The SCHEMA option may not be used with AS DIF or AS MPLAN (468)

You attempted to use UNLOAD SCHEMA with an AS DIF or AS MPLAN clause.

The USER and OWNER passwords must match (303)

Your USER and OWNER passwords must be the same to use the UNLOAD command. Set your USER password to match your OWNER password to continue.

Unidentified AS clause (302)

The AS clause specified something other than ASCII, DIF or MPLAN.
USER

Illegal password name (276)

You have attempted to enter an invalid USER password. Passwords must be from 1 to 8 characters long.

WHILE...THEN...ENDWHILE

ENDWHILE or ENDIF missing in an input file (462)

Your command file does not contain a matching ENDWHILE command for a WHILE.
Check your command file. See the "Command Dictionary" for information.

Illegal Boolean operator between variable and value (22)

The operator between a variable and a value was not a recognized boolean operator.
Valid operators are: EXISTS, FAILS, EQ, NE, GT, GE, LT, LE, CONTAINS, EQA, NEA, GTA, GEA, LTA, LEA, =, >, <, <=, >=, and <>.

No matching ENDIF or ENDWHILE found--Check command file code (464)

Your command file attempted to exit a WHILE...ENDWHILE structure by executing a GOTO command that referred to a LABEL outside of the WHILE...ENDWHILE structure. Restructure your command file logic so that all WHILE...ENDWHILE structures terminate only by execution of the ENDWHILE command.

No WHILE or IF commands are open on current input source (466)

R:base has found an ENDWHILE statement with no matching WHILE.

Too many commands in WHILE loop (218)

All commands from the start of a WHILE loop to the corresponding ENDWHILE command are stored in a scratch buffer. This buffer is limited to 4000 characters and you have exceeded this limit.

Too many nested WHILE commands (217)

You cannot nest WHILE conditions more than 10 levels deep.

WHILE and IF commands are nested beyond maximum (465)

You may have a maximum of 10 nested WHILE loops in a command file and you have exceeded this number. Restructure your command file.

WRITE

Column is out of bounds (417)

Col is out of bounds (562)

The specified screen column in the AT clause is out of range. The screen column may be 1 to 79.

Row is out of bounds (561)

Row out of bounds (415)

The specified screen row in the AT clause is out of range. The screen row may be 1 to 24.

RCOMPILE Error Messages

Duplicate name - overwrite (Y or N):

You have entered a duplicate name. Press [Y] to overwrite the existing file. Press [N] to enter a new name.

-ERROR- Directory file full

The procedure directory already has 42 entries. You cannot add another compiled file to this procedure file.

-ERROR- Illegal filename

The file name you specified does not conform to the file naming rules. A file name must be 1-8 characters long and may have a 1-3 character extension.

-ERROR- I/O Problems - check for a full disk

RCOMPILE cannot write a file for some reason. Either the disk is full or corrupt. Use CHKDSK to determine either condition.

-ERROR- Illegal DOS path name

You have tried to use a path that RCOMPILE cannot handle. See if you have entered the path correctly.

ERROR- Incomplete EXPRESS file

When reading a file created by EXPRESS, RCOMPILE found an end of file marker before the EXPRESS command structure was complete. The file may have been truncated or damaged. Use the EXPRESS to check the completeness of application files.

-ERROR- Invalid menu type

Menus must be either ROW or COLUMN. RCOMPILE has encountered a menu with a definition either than ROW or COLUMN.

-ERROR- Too much menu text

RCOMPILE has encountered a menu which contains more text than is allowed.

-ERROR- Unable to open file

RCOMPILE was unable to open the specified file. Check that you correctly spelled the file name and specified the drive and directory, if needed.

More than 1600 bytes on record

When reading a command file, RCOMPILE found an entry with more than 1600 bytes of information. Check the command file for lines that are longer than 1600 characters.

See documentation for help

You have pressed the [F10] function key which is a request for on-line help. RCOMPILE does not currently support on-line documentation. See chapter 15 for information.

FileGateway Error Messages

Ambiguity in fields. Return to field definition? (Y/N)

Ambiguous field definition.

Mismatched start and end markers. Make sure that the S and E used when marking ASCII fixed field markers match. A single S may be used to mark a one-character field.

Column name <colname> already exists in this table. Please rename the field.

The field name appears more than once in the PFS:FILE file. Enter a different field name.

<colname> is not a legal column name.

The column name you have entered is not valid. Enter a new column name 1 to 8 characters long.

Coordinates specified incorrectly. Please reenter.

The SYLK file coordinates were not entered properly. Reenter the coordinates in the form R#C#.

Database contains additional password protected tables.

You attempted to display the structure of a database containing additional tables with password protection. Use R:base to remove password protection from the tables.

Delimiter not found in first record. Please recheck the input file.

The specified ASCII delimiter could not be found. Check that you entered the correct file name. Check that you entered the correct delimiter for the input file.

Error in closing table - <tblname>.

Error in writing table - <tblname>.

The FileGateway could not close or write one or more of the database tables. Restart the conversion from the beginning. Make sure there is enough room on the disk for the database and that the disk is not defective using CHKDSK.

Error in writing to exception file.

The FileGateway could not write the exceptions to the designated exception file. Check for disk error conditions with CHKDSK.

Error reading ASCII input file.

The FileGateway could not read the specified ASCII file. Check the file name. If it is correct, check the integrity of the file by using the Help options.

Error reading dBASE II file.

The FileGateway could not read the specified dBASE II file. Check the file name. If it is correct, check the integrity of the file using the dBASE II software.

Error reading Lotus file.

The FileGateway could not read the specified Lotus file. Make sure the file name is correct including the .WKS extension.

<fieldname> already exists in file as type <datatype> .

You have assigned a column (field) name which is identical to an existing column name in the database but the data types do not match. You must rename the current column or change the data type to match the existing column.

File not found.

FileGateway could not locate the requested input file. Make sure that the file name was spelled correctly and that any required drive designation and path was included. Occurs when attempting load, type, or display the directory.

File probably not a dBASE II file.

The specified dBASE II file could not be read. Check to see if you are using a file from a product other than dBASE II. Also, check to see if you entered the file name correctly.

Illegal column name.

The column name you entered violated the naming rules. Column names must be 1 to 8 characters.

Illegal database name.

The database name you entered violated the naming rules. Database names must 1 to 7 characters.

Illegal path.

You have specified an illegal path name. Be sure you follow DOS path rules.

Input file has too many fields to add to database.

The input file has more than 400 fields. Use the original file product to split the input file into smaller segments.

I/O ERROR in writing database table.

The FileGateway could not write the converted table to the disk. This usually indicates a problem with the floppy disk. Make sure you have a disk in the specified disk drive. Use CHKDSK to validate the disk.

Label fields could not be found.

The FileGateway could not find the field labels in the specified Lotus coordinate area.

More than 1600 characters of data on record.

The ASCII input records in a file are longer than 1600 characters. If the input records are longer than 1600 characters, the file cannot be converted by the FileGateway. Shorten the input records, if possible.

No data available in the specified area.

The FileGateway could not find data in the Lotus data area specified by the coordinates you entered. Check that the coordinates are correct for the data area.

No more tables can be added to this database. Maximum number is 40.

The database you have specified already contains 40 tables which is the maximum allowed. Create a new database to hold the new table or use the REMOVE command to delete unneeded tables from the database.

Not enough disk space for converted file on the database disk.

Choose a different disk for the database.

There is insufficient room on the specified data disk for the file. Exit to DOS to delete unnecessary files from the data disk. Or, copy the database files to another disk. Then, restart the FileGateway.

Number of fields defined differs from existing table.

In attempting to add input file data to an existing database table, the FileGateway has found that the number of fields in the input file does not match the number of columns in the existing table. Create a new table for the input data.

Out of file handles.

You must have a FILES=20 line in your CONFIG.SYS file. See chapter 1 for installation and start-up instructions.

Path not found.

The path name you specified is not on the disk. Check the spelling of the path name. Are you certain the data disk has the specified path? See your DOS Manual for information on paths.

Please enter a name of eight characters or less or [F10] for help:

You have entered a table name that is not valid. Enter a valid 1 to 8 character table name.

Range specification not found in file.

The data range specified was not found in the SYLK worksheet file. Check the coordinates of the worksheet. Make sure that data is present within the given coordinates.

Range specified incorrectly. Use standard Lotus format.

The coordinates you have specified do not match standard Lotus format. Use standard format entering the worksheet upper-left and lower-right coordinates, for example A1...E20.

Record # <recnum> too long. It will be skipped.

The PFS:FILE record you are attempting to convert has more than 1600 characters. Separate the PFS:FILE worksheet into more than one file to make records shorter.

The following label in your PFS:FILE form is too long to be used as a column name.

The PFS:FILE field name is too long. Enter a new field name 1 to 8 characters long.

There are no tables in this database.

This message is displayed when requesting the database structure from the Help Menu and you have not completed loading the first file into the database. Complete the loading process then recheck the database structure from the help menu.

The number of columns in a database cannot exceed 400.

You have attempted to define more than 400 columns for the database. Decide what information must be grouped together and create a second database.

Unable to add data to an existing table.

This message can occur for a number of reasons. Specifically, it is displayed when you are attempting to add data to an existing table in a database. The reason may be that the number of fields in the input file does not match the number of columns in the table or that data types in the input fields do not match the column data types, or for any incompatibility that the FileGateway can discern between the input file and the existing table.

Unable to convert file. Error in SYLK file 'B:' record.

Unable to convert file. No 'B:' record in SYLK file.

The SYLK file's B: record was not readable because a) it was bad, or b) it was not there. Make sure you specified the correct file name. The SYLK file may not have transferred properly from the original Multiplan file.

Unable to convert file. Not recognized as DIF format.

The file you are trying to convert cannot be converted as a DIF file. Make sure the original file is in DIF format. Make sure the file name is the DIF file not the original VisiCalc or other program's file.

Unable to define a database from the input data.

The FileGateway could not create a database from the original file data. See any accompanying message for the specific problem. If none, then retry the option. Make sure the file you want is on the designated drive. This message also occurs when the information is too large. For example, a Lotus file has too many fields or an ASCII file record has no RETURN or records that are too long.

Unable to open database.

Unable to open file.

Unable to open R:base file.

Unable to open this file. Check file name or path.

Unable to open R:base database table.

The FileGateway could not open the requested database, table, or input file. Check the directory and provide paths, if necessary.

Unable to open exception file.

The FileGateway cannot open the exception file (EXCEPT.DAT). Make sure there is enough room on the disk for the exception file.

Unable to read file. Only ASCII files can be listed.

You tried to list a non-ASCII file from the Help Menu. Make sure the requested file is an ASCII file.

Unable to read PFS input file.

The FileGateway could not read the PFS:FILE input file. Make sure you specified the correct PFS:FILE file name.

Unrecognized input file.

The specified input file could not be read. Check that you entered the correct file name.

User not authorized on any of the existing tables.

All of the existing tables in the requested database are password protected. You must use R:base to remove password protection from the tables you want to use.

EXPRESS Error Messages

GENERAL EXPRESS ERROR MESSAGES

-ERROR- There are no databases created

There are no databases defined on the default subdirectory. Use the COPY command to copy the database you want to use onto the default subdirectory or use the CHDIR command to change the default to the subdirectory that contains the database.

-ERROR- Unable to open the requested database

The EXPRESS could not open the specified database. Verify that you spelled the database name correctly

-ERROR- Unauthorized database access

You attempted to open a database that is OWNER password protected. You cannot open the database unless you know the password.

-ERROR- Owner password does not match database

The database you are defining has an owner password that does not match the password you entered. Check to see if you entered the password correctly.

EXPRESS help file not found

The EXPRESS searched for the EXPRESS.HLP file and could not find it. Check your disk to make sure that you copied the file correctly.

A WHERE clause is required for delete

This message tells you that a WHERE clause is required for the delete option. The purpose of this WHERE clause is to prevent inadvertent global changes.

The requested "macro" file does not exist

You typed in the name of a command file that the EXPRESS does not recognize. Check to make sure that you typed in the name of the file correctly.

File not found

The EXPRESS could not find the specified file. Verify that you spelled the file name correctly.

R:BASE.DAT exists - Do you want to overwrite it?

The RBASE.DAT file already exists on the default directory. If you want to keep the file, select NO.

-ERROR- Directory file full

The procedure directory already has 42 entries. You cannot add another compiled file to this procedure file.

-ERROR- I/O Problems - check for a full disk

Verify the quantity of free space on the disk. If your disk is full, delete unneeded files or use the PACK command to release unused space.

DATABASE DEFINITION ERROR MESSAGES

-ERROR- Illegal database name

Database names must be 1-7 characters. Blank spaces are not recognized as characters.

-ERROR- The requested database already exists

You attempted to define a database that already exists. If you want to define a new database, you must give it a unique name.

No tables are defined for this database

You attempted to list the table names in the database and received this message which indicates that you have either removed all the tables, or have not defined any tables.

No columns are defined for this table

You attempted to list columns in a table where no columns are yet defined.

Table name is a duplicate

You attempted to define a table which already exists in the database. Use a new table name.

Reserved words cannot be used for table names

You attempted to name a table using a reserved word. Choose a name that is not a reserved word. For a list of reserved words, see chapter 5.

This table contains the limit of 40 columns

You cannot define more than 40 columns for a given table when using the EXPRESS. If you want to use the EXPRESS, you must first delete columns that you do not need or reorganize the structure of the table. If you must have more than 40 columns in a table, use the R:base DEFINE mode to define your database.

Too many tables defined - limit is 40

You may define no more than 40 tables for a database. Use the REMOVE command to delete unneeded tables.

Too many column names defined – limit is 400

The full capacity of R:base is 400 columns in one database. You must remove one or more columns from the database before you can define a new column.

Column name is a duplicate

The specified column name is used more than once in the same table. Use the LIST COLUMNS command to display a list of columns defined for the table and choose a unique name for the new column you are defining.

Reserved words cannot be used for column names

You attempted to name a column using a reserved word. Choose a name that is not a reserved word. For a list of reserved words, see chapter 5.

Tables with data may not be removed

Before you can remove a table, you must first delete the data in the table. Use the DELETE ROWS command to do this.

Only the first 40 column names are shown

The display of column names in the EXPRESS is limited to 40. Therefore, you cannot define more than 40 columns for each table.

Table width exceeds 1530 characters

The maximum length of a row is 1530 characters and the table you are attempting to add will make the row longer than 1530 characters. Remove any unneeded columns from the table.

Column width exceeds 1500 characters

The maximum length specification for a TEXT column in a column definition is 1500 characters. The minimum specification is one character.

FORMS ERROR MESSAGES

The limit of 40 forms has been reached

You are allowed to define up to 40 forms. Remove unneeded forms before defining another.

-ERROR- Form does not match table

You have selected a form that was defined for a table other than the table you specified. Select a different form or define a new one.

Too many tables defined – limit is 40

FORMS is an R:base table. If 40 tables exist in a database, you cannot define forms unless you remove a table.

Too many column names defined – limit is 400

FORMS uses two column names, FNAME and FDATA. If your database already has 400 columns, you cannot define forms until you remove two columns. Delete unused column names or tables.

Form name is a duplicate

The form name you chose is the name of an already existing form. Choose a unique name for the new form.

REPORTS ERROR MESSAGES

The limit of 40 reports has been reached

You are allowed to define up to 40 reports. Remove unused reports before defining another.

-ERROR- Report does not match table

You selected a report defined for a table other than the table you specified. Select a different report or define a new one.

Too many tables defined – limit is 40

REPORTS is an R:base table. If 40 tables exist in a database, you cannot define a report unless you remove a table.

Too many column names defined – limit is 400

REPORTS uses two column names. If your database already has 400 columns, you cannot define a report until you remove two columns. Delete unused column names or tables.

Report name is a duplicate

The report name you chose is the name of an already existing report. Choose a unique name for the new report.

The maximum page width has been reached

While specifying column names to include in a report, the sum of the specified column widths exceeded 132, the maximum page width for reports. You need to remove a column from the report.

APPLICATION ERROR MESSAGES

-ERROR- There are no applications created

There are no applications defined on the default subdirectory. Use the COPY command to copy the application you want to use onto the default subdirectory or use the CHDIR command to change the default to the subdirectory that contains the application.

-ERROR- The requested application already exists

You are attempting to define an application that already exists. Select a unique name for the new application.

Too many menus defined

You can only define up to 15 menus. If you need to define another menu, you must first delete menus that you do not need.

Menu name is a duplicate

You attempted to assign an existing help, custom, or macro menu name to the menu you are defining. Check the list of names and assign a unique name to the new menu.

There are no menus defined for this application

There are no menus defined for the application you are currently using.

There are no actions defined for this menu

There are no actions defined for the menu you are currently using or editing.

Text name is duplicated

You attempted to assign an existing menu name to the menu text you are defining. Check the list of names and assign a unique name to the new text.

This menu contains the limit of 12 choices

You have exceeded the 12 menu options limit for horizontal menu. If you wish to add a new option, you must first delete one that you do not need.

This menu contains the limit of 9 choices

You have exceeded the 9 menu options limit for a vertical menu. If you wish to add a new option, you must first delete one that you do not need.