

RETURN TO MAIN MENU

Recital/4GL Set Commands

Recital/4GL Set Commands

Recital Corporation,
100 Cummings Center, Suite 318J
Beverly, MA 01915

Recital may have patents and/or patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.
COPYRIGHT ©1988-2006 Recital Corporation. All rights reserved. All Recital products are trademarks or registered trademarks of Recital Corporation, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.

Last Updated June, 2006

INDEX

SET ALTERNATE	1
SET ASSERTS	3
SET AUTOCATALOG	4
SET AUTOFORMAT	5
SET AUTOJOIN	6
SET AUTOREPAIR	7
SET AUTOSAVE	8
SET BELL	9
SET BLINK	10
SET BLOCKSIZE	11
SET BORDER	12
SET CACHELOAD	13
SET CAPTURE	14
SET CARRY	15
SET CATALOG	16
SET CENTURY	17
SET CLASSLIBRARY	18
SET CLEAR	19
SET CLIPPER	20
SET CLIPPER5	22
SET CLOCK	24
SET CLOCKDISPLAY	25
SET CLOCKRATE	26
SET COLOR	27
SET COLOR OF	29
SET COMMAND	30
SET COMMANDWINDOW	31
SET COMPATIBLE	32
SET COMPILE	37
SET CONFIRM	38
SET CONSOLE	39
SET CURRENCY	40
SET CURSOR	41
SET DATE	42
SET DBTRAP	43
SET DCACHE	44
SET DEBUG	45
SET DECIMALS	46
SET DEFAULT	47
SET DELETED	48
SET DELIMITERS	49
SET DESIGN	51
SET DESCRIPTIONS	50
SET DEVELOPMENT	52
SET DEVICE	53
SET DIALOG	54
SET DICTIONARY	55
SET DIRECTORY	56
SET DISPLAY	57

SET DOESCAPE	58
SET DOHISTORY	59
SET ECHO	60
SET EDITFIELD	61
SET EMACROS	62
SET ENCRYPTION	63
SET EPOCH	64
SET ERRORVERSION	65
SET ERRORWINDOW	66
SET ESCAPE	67
SET EXACT	68
SET EXCLUSIVE	69
SET FASTINDEX	70
SET FCACHE	71
SET FIELDS	72
SET FIELDVAL	73
SET FILECASE	74
SET FILETYPE	75
SET FILTER	77
SET FIXED	78
SET FKLABEL	79
SET FORMAT	80
SET FORMSTATE	81
SET FORMUPDATE	82
SET FULLPATH	83
SET FUNCTION	84
SET GATEWAY	85
SET GCACHE	86
SET HEADING	87
SET HELP	88
SET HELPFILE	89
SET HELPWINDOW	90
SET HIDDENFIELD	91
SET HIGHLIGHT	92
SET HISTORY	93
SET HISTPATH	95
SET HOURS	96
SET IBLOCK	97
SET ICACHE	98
SET INDEX	99
SET INDEXEXT	100
SET INKEYDELAY	101
SET INSTRUCT	102
SET INTENSITY	103
SET JOURNAL	104
SET KBEDIT	106
SET KEY	107
SET KEY...TO	108
SET KEYWORD	109
SET LANGUAGE	110
SET LDCHECK	111
SET LIBRARY	112
SET LOCK[WAIT]	113

SET LOCKTYPE	114
SET MACKEY	115
SET MACROS	116
SET MAIL	117
SET MARGIN	119
SET MAPCHAR	118
SET MARK	120
SET MAXDBO	121
SET MBLOCK	122
SET MCONFIRM	123
SET MEMOCLEAR	124
SET MEMOEXT	125
SET MEMOFORMAT	126
SET MEMOJOURNAL	127
SET MEMOSOFTCR	128
SET MEMOWIDTH	129
SET MEMOWINDOW	130
SET MENU	131
SET MENUBAR	132
SET MESSAGE	133
SET MOUSE	134
SET MULTIUSER	135
SET NAVIGATE	136
SET NEAR	137
SET NOTIFY	138
SET NULL	139
SET NULLDISPLAY	141
SET ODOMETER	142
SET ONERROR	143
SET OPTIMIZE	144
SET OPTLOG	145
SET ORDER	146
SET PAGELENGTH	147
SET PAGEWIDTH	148
SET PATH	149
SET PAUSE	150
SET PCACHE	151
SET PCEDIT	152
SET PCEXACT	153
SET PCFILTER	154
SET PCFKEYS	155
SET PCGRAPHICS	156
SET PCKEYS	157
SET PCLOCKING	159
SET PCPICTURE	160
SET PCSAYS	161
SET PCUNIQUE	162
SET PERFDIAL	163
SET PERFMETER	164
SET POINT	165
SET POSTFORM	166
SET POSTMENU	167
SET POSTRECORD	168

SET PRECISION	170
SET PREFORM	171
SET PREMENU	172
SET PRERECORD	173
SET PRINT	174
SET PRINTER	175
SET PROCEDURE	176
SET PROMPT	177
SET PSHARE	178
SET QUERYMODE	179
SET READEXIT	180
SET READINSERT	181
SET REFRESH	182
SET RELATION	183
SET REPROCESS	185
SET RETAINMENU	186
SET ROLLBACK	187
SET RUNCLEAR	188
SET RUNWAIT	189
SET SAFETY	190
SET SCHEDULE	191
SET SCOREBOARD	193
SET SCREENCAPTURE	194
SET SCREENIO	196
SET SCREENMAP	197
SET SCROLL	199
SET SECONDS	200
SET SEPARATOR	201
SET SEQNO	202
SET SHADOW	203
SET SKIP	204
SET SOFTSEEK	205
SET SPACE	206
SET SQL	207
SET SQLDIALECT	208
SET SQLPROMPT	209
SET SQLROWID	210
SET STATUS	211
SET STEP	212
SET STRINGBUF	213
SET STRINGLEN	214
SET SYSLOGGING	215
SET SYSMENU	216
SET TALK	217
SET TBUFSIZE	218
SET TCACHE	219
SET TEDIT	221
SET TERMINAL	222
SET TEXTMERGE	223
SET TIME	225
SET TIMEOUT	226
SET TITLE	227
SET TMPDIR	228

SET TRACEWINDOW	229
SET TRANSACTION	230
SET TRAP	232
SET TSLENGTH	233
SET TYPEAHEAD	234
SET UDFPARMS	235
SET UICONTROLS	236
SET UNDERLINE	237
SET UNIQUE	238
SET UPDATE	239
SET VALIDATE	240
SET VAXTIME	241
SET VERIFY	242
SET VIEW	243
SET WINDOW OF BROWSE	244
SET WINDOW OF EDIT	245
SET WINDOW OF MEMO	246
SET WINDOW OF RELATION	247
SET WP	249
SET WRAP	250
SET XMLFORMAT	251

Set Commands by Category

Databases

SET AUTOCATALOG		
-----------------	--	--

Date and Time Data

SET CENTURY	SET DATE	SET EPOCH
SET MARK	SET SECONDS	SET VAXTIME

DES3 Encryption

SET ENCRYPTION		
----------------	--	--

Environment

SET CLASSLIBRARY	SET COMPILE	SET DEFAULT
SET DEVELOPMENT	SET DESCRIPTIONS	SET DIRECTORY
SET EMACROS	SET EXACT	SET EXCLUSIVE
SET FILECASE	SET FULLPATH	SET LIBRARY
SET LOCKWAIT	SET MACROS	SET MAXDBO
SET MULTIUSER	SET OPTLOG	SET PROCEDURE
SET SAFETY	SET SQL	SET STRINGBUF
SET STRINGLEN	SET SYSLOGGING	SET TMPDIR
SET UDFPARMS		

Error Handling and Debugging

SET ASSERTS	SET DEBUG	SET DOHISTORY
SET ECHO	SET ERRORVERSION	SET HISTORY
SET HISTPATH	SET ONERROR	SET STEP

Fields and Records

SET DELETED	SET FIELDS	SET HIDDENFIELD
-------------	------------	-----------------

Indexing

SET AUTOJOIN	SET AUTOREPAIR	SET FASTINDEX
SET INDEX	SET NEAR	SET ORDER
SET SOFTSEEK	SET UNIQUE	

Information Center

SET CATALOG	SET DESIGN	SET OPTIMIZE
SET TITLE		

Input/Output

SET ALTERNATE	SET DEVICE	SET HEADING
SET MAPCHAR	SET SPACE	SET TEXTMERGE
SET UICONTROLS		

Keyboard Events

SET CAPTURE	SET CURSOR	SET DOESCAPE
SET ESCAPE	SET INKEYDELAY	SET KBEDIT
SET KEY	SET KEY...TO	SET MACKEY
SET PCFKEYS	SET TYPEAHEAD	

Manual Locking

SET LOCKTYPE		
--------------	--	--

Memos

SET MEMOEXT	SET MEMOFORMAT	SET MEMOJOURNAL
SET MEMOSOFTCR	SET MEMOWIDTH	

Numeric Data

SET CURRENCY	SET DECIMALS	SET FIXED
SET POINT	SET SEPARATOR	

Performance and Optimization

SET CACHELOAD	SET DCACHE	SET FCACHE
SET GCACHE	SET ICACHE	SET PCACHE
SET PSHARE		

Printing

SET MARGIN	SET PAGELength	SET PAGEWIDTH
SET PRINT	SET PRINTER	

Recital Terminal Developer Environment

SET CLEAR	SET CLOCK	SET CLOCKDISPLAY
SET CLOCKRATE	SET COMMAND	SET FUNCTION
SET HELPFILe	SET HIGHLIGHT	SET HOURS
SET INSTRUCT	SET MAIL	SET MEMOCLEAR
SET NOTIFY	SET ODOMETER	SET PAUSE
SET PERFDIAL	SET PERFMETER	SET PROMPT
SET RUNCLEAR	SET RUNWAIT	SET SCOREBOARD
SET SCREENIO	SET SQLPROMPT	SET STATUS
SET SYSMENU	SET TALK	SET TBUFSIZE
SET TEDIT	SET TERMINAL	SET TIME
SET WP		

Screen Environment

SET COLOR	SET COLOR OF	SET CONFIRM
SET CONSOLE	SET DIALOG	SET KEYWORD
SET LANGUAGE	SET MCONFIRM	SET MEMOWINDOW
SET MESSAGE	SET RETAINMENU	
SET SCREENMAP	SET SCROLL	SET SHADOW
SET TIMEOUT	SET VERIFY	SET WRAP

Screen Forms

SET AUTOFORMAT	SET AUTOSAVE	SET BELL
SET BLINK	SET BORDER	SET CARRY
SET DELIMITERS	SET EDITFIELD	SET FIELDVAL
SET FKLABEL	SET FORMAT	SET FORMSTATE
SET FORMUPDATE	SET HELP	SET INTENSITY
SET MENU	SET MENUBAR	SET MOUSE
SET NAVIGATE	SET POSTFORM	SET POSTMENU
SET POSTRECORD	SET PREFORM	SET PREMENU
SET PRERECORD	SET QUERYMODE	SET READEXIT
SET READINSERT	SET SCHEDULE	SET SCREENCAPTURE
SET UNDERLINE	SET UPDATE	SET VALIDATE

Screen Windows

SET COMMANDWINDOW	SET ERRORWINDOW	SET HELPWINDOW
SET TRACEWINDOW	SET WINDOW OF BROWSE	SET WINDOW OF EDIT
SET WINDOW OF MEMO	SET WINDOW OF RELATION	

SQL Applications

SET NULL	SET NULLDISPLAY	SET SQLDIALECT
SET SQLROWID	SET TCACHE	SET XMLFORMAT

Table Organization

SET DICTIONARY	SET FILTER	SET GATEWAY
SET JOURNAL	SET RELATION	SET ROLLBACK
SET SEQNO	SET SKIP	SET TRANSACTION
SET VIEW		

Text Searching

SET TSLENGTH		
--------------	--	--

Xbase Compatibility

SET BLOCKSIZE	SET CLIPPER	SET CLIPPER5
SET COMPATIBLE	SET DBTRAP	SET DISPLAY
SET FILETYPE	SET IBLOCK	SET INDEXEXT
SET LDCHECK	SET MBLOCK	SET PCEDIT
SET PCEXACT	SET PCFILTER	SET PCGRAPHICS
SET PCKEYS	SET PCLOCKING	SET PCPICTURE
SET PCSAYS	SET PCUNIQUE	SET PRECISION
SET REFRESH	SET REPROCESS	SET TRAP

SET ALTERNATE

Class

Input/Output

Purpose

Capture output text in a file

Syntax

SET ALTERNATE ON | OFF | (<expL>)

SET ALTERNATE TO [<.txt filename> | (<expC>)]

See Also

CLOSE ALTERNATE, SET DEVICE, PRINT, SET PRINT, SET PRINTER

Description

The SET ALTERNATE command provides the ability to record all textual output into a file. This command is primarily used when generating reports, which can later be printed on the system printer with the PRINT command. Full screen form operations are not recorded in the alternate file, but @...SAY commands are.

The SET ALTERNATE TO command is used to specify the target filename for the operation. If no filename extension is specified, then '.txt' is used. The filename can be substituted with an <expC>, enclosed in round brackets, which returns a valid filename.

You may enable and disable the writing of output to the alternate file with the SET ALTERNATE ON and OFF command. To close the alternate file use either the CLOSE ALTERNATE command or issue a SET ALTERNATE TO with no filename specified.

If SET ALTERNATE ON is issued, and no SET ALTERNATE TO <.txt filename> is active, then the alternate filename 'alternate.txt' will be used. This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, ALTERNATE is set ON. If a value of .F. is returned, ALTERNATE is set OFF. Note that <expL> should be enclosed in round brackets.

By default, ALTERNATE is OFF.

Example

```
set alternate to alt
? [Write this text to file]
set alternate off
? [Don't write this text to file]
?
set console off
set screenmap off
set device to print
set alternate on
setprc(0,0)      && set coordinates to 0,0
@1,0 say [write this to file to the file, not to the screen]
close alternate
set console on
set screenmap on
set device to screen
return
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ASSERTS

Class

Error Handling and Debugging

Purpose

Enable or disable the display of assert message dialogs

Syntax

SET ASSERTS ON | OFF | (<expL>)

See Also

ASSERT, DEBUG

Description

The SET ASSERTS command is used to enable or disable the display of assert message dialogs. Assert messages are used for program debugging purposes. The ASSERT command checks a specified condition and displays a message dialog if the condition evaluates to False (.F.). The dialog provides options to Debug, Cancel, Ignore and Ignore All.

When SET ASSERTS is ON, the display of assert message dialogs is enabled. When SET ASSERTS is OFF, the display of assert message dialogs is disabled. By default, SET ASSERTS is OFF.

Example

```
set asserts on
parameters para1, para2, para3
assert pcount() = 3 message [3 parameters required]
// code continues
return
```

Products

Recital Mirage Server, Recital Terminal Developer

SET AUTOCATALOG

Class

Databases

Purpose

Enable files to be automatically added to a database catalog

Syntax

```
SET AUTOCATALOG ON | OFF | (<expL>)  
SET AUTOCATALOG TO <database>
```

See Also

ALTER INDEX, ALTER TABLE, CLOSE DATABASES, CLOSE TABLES, COMPILE DATABASE, CREATE DATABASE, CREATE INDEX, CREATE TABLE, CREATE VIEW, DISPLAY DATABASE, DISPLAY INDEXES, DISPLAY TABLES, DROP DATABASE, DROP INDEX, DROP TABLE, LIST DATABASE, LIST INDEXES, LIST TABLES, OPEN DATABASE, PACK DATABASE, REBUILD DATABASE, REINDEX DATABASE, USE, SET EXCLUSIVE, ADATABASES(), DBUSED(), DB_MAXWKA

Description

The SET AUTOCATALOG commands allow tables and their index files to be automatically added to a database catalog. The database itself should be closed when using the auto catalog commands. The SET AUTOCATALOG TO <database> command specifies the name of the database for which the catalog should be updated. SET AUTOCATALOG ON | OFF allows updates to the catalog to be toggled on and off. Once the auto catalog commands are active, tables and their indexes can be opened from the interactive prompt or from an application and the database catalog will automatically be updated. For additional details on the information stored in the database catalog, please see the OPEN DATABASE command.

Databases in Recital are implemented as directories containing files that correspond to the tables and associated files in the database. Operating System file protection can be applied individually to the files for added security. The directories are sub-directories of the Recital data directory. The environment variable / symbol DB_DATADIR points to the current Recital data directory. Databases are opened using the OPEN DATABASE command. Database commands can be issued as SQL statement commands or as Recital/4GL commands.

Example

```
close databases  
set autocatalog to southwind  
set autocatalog on  
do myapp
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET AUTOFORMAT

Class

Screen Forms

Purpose

Enable automatic screen format file opening

Syntax

SET AUTOFORMAT ON | OFF | (<expL>)

See Also

APPEND, CHANGE, EDIT, INSERT, QUERY, USE, FMT(), SET FORMAT

Description

The SET AUTOFORMAT command is used to enable or disable automatic screen format file opening. If SET AUTOFORMAT is ON, an available screen format file (.fmt) with the same basename as the table will be opened automatically when the table is opened. If SET AUTOFORMAT is OFF, screen format files must be opened explicitly with the SET FORMAT command.

This command allows the optional <expL> to be evaluated. If a value of .T. is returned, AUTOFORMAT is set ON, if a value of .F. is returned, AUTOFORMAT is set OFF. Note that <expL> should be enclosed in round brackets. By default, SET AUTOFORMAT is ON.

Example

```
set autoformat on
use product.rdb
? fmt()
product.fmt
```

Products

Recital Mirage Server, Recital Terminal Developer

SET AUTOJOIN

Class

Indexing

Purpose

Enable automatic creation of indexes to satisfy relationships

Syntax

SET AUTOJOIN ON | OFF | (<expL>)

See Also

SET RELATION, SET VIEW, INDEX, REINDEX

Description

If SET AUTOJOIN is ON, then an index file is automatically generated to satisfy the relationship between two tables when a SET RELATION TO command is executed.

When the SET RELATION TO command is executed, a check is made to see if the target table which is being related has an index associated with it. If not, a temporary index is generated on a field in the target table which has the same name, data type and width as that which is specified in the SET RELATION command. The index is a single index file with a '.tmp' extension. If no matching field exists, an error is given.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, AUTOJOIN is set ON. If a value of .F. is returned, AUTOJOIN is set OFF. Note that <expL> should be enclosed in round brackets. You should create the required index files to satisfy any permanent relationships, so that they are not created each time the relationship is specified. The use of AUTOJOIN should be limited to temporary views across tables only.

By default, AUTOJOIN is OFF.

Example

```
select a
use addresses
select b
use patrons
set autojoin on
set relation to name into addresses
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET AUTOREPAIR

Class

Indexing

Purpose

Repair indexes on the fly

Syntax

SET AUTOREPAIR ON|OFF(<expL>)

See Also

INDEX, REINDEX

Description

The SET AUTOREPAIR command allows enhanced management of database indexes. When SET AUTOREPAIR is ON the Recital DBMS will automatically attempt to patch a corrupt index file whenever the *Record Not In Index* error message is generated. All database and index caches for the user are flushed, the record is locked, and the index file searched for missing or invalid keys related to the active record. If the index entry is missing, the entry is added to the index file. If the index entry is invalid, the key is updated based on index key calculations for the current record. All active index files are updated for the current record number only.

This command should only be used by database administrators with full working knowledge of the application and the proper operation of functional indexes. The use of AUTOREPAIR is meant to aid in patching index files without the overhead of shutting down running applications and rebuilding index files in exclusive mode.

The source of all *Record Not In Index* errors should be investigated and steps must be taken to resolve the source of the problem. Not all *Record Not In Index* error messages indicate problems in index files as some application algorithms may use temporary index files built on limited data sets.

Example

set autorepair on

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET AUTOSAVE

Class

Screen Forms

Purpose

Save to disk after every change

Syntax

SET AUTOSAVE ON | OFF | (<expL>)

See Also

CHANGE, EDIT, FFLUSH(), SAVE GETS

Description

The SET AUTOSAVE command is used to enable or disable automatic saving of records to disk after every change users make. When AUTOSAVE is set ON, changes to records are saved to disk immediately following an I/O operation. The SET AUTOSAVE command has no effect in APPEND mode, where changes are saved when the [NEXT RECORD] and/or [EXIT SAVE] keys are pressed. When AUTOSAVE is OFF, pressing the [ABANDON] key reverses changes made to records. Changes to records are saved to disk when the [EXIT/SAVE], [NEXT/SCREEN], [PREVIOUS SCREEN] or [MENUBAR] keys are pressed.

This command allows the optional <expL> to be evaluated. If a value of .T. is returned, AUTOSAVE is set ON, if a value of .F. is returned, AUTOSAVE is set OFF. Note that <expL> should be enclosed in round brackets. By default, SET AUTOSAVE is OFF.

Example

```
set autosave on
```

Products

Recital Mirage Server, Recital Terminal Developer

SET BELL

Class

Screen Forms

Purpose

Enable or disable the terminal bell inside forms

Syntax

SET BELL ON | OFF | (<expL>)

See Also

SET CONFIRM, CHR(), @...GET, TONE()

Description

If SET BELL is ON then the terminal bell will sound if an attempt is made to enter invalid data (e.g. alphabetic character in a numeric field), or enter more characters into a field than it can contain when SET CONFIRM is ON. This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, BELL is set ON. If a value of .F. is returned, BELL is set OFF. Note that <expL> should be enclosed in round brackets. By default, BELL is ON.

Example

```
set bell off
use patrons
edit
```

Products

Recital Mirage Server, Recital Terminal Developer

SET BLINK

Class

Screen Forms

Purpose

Enable or disable blinking of field contents when help is active

Syntax

SET BLINK ON | OFF | (<expL>)

See Also

SET HIGHLIGHT, SET INTENSITY, SET UNDERLINE, SET DELIMITERS, @...GET, READ, EDIT, BROWSE, APPEND, BLINK(), BOLD(), REVERSE(), UNDERLINE()

Description

The SET BLINK ON command enables the associated field to blink when a user presses the [HELP] key to obtain a pop-up menu. This has been added as an enhancement to your user work surfaces. This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, BLINK is set ON. If a value of .F. is returned, BLINK is set OFF. Note that the <expL> should be enclosed in round brackets. The default setting for the BLINK command is OFF.

Example

set blink on

Products

Recital Terminal Developer

SET BLOCKSIZE

Class

Xbase Compatibility

Purpose

Changes the default block size of memo fields

Syntax

SET BLOCKSIZE TO <expN>

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT

Description

This command changes the block size of dBASE IV and FoxPro memo fields.

Example

set blocksize to 3

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET BORDER

Class

Screen Forms

Purpose

Changes bordering

Syntax

SET BORDER TO SINGLE | DOUBLE | PANEL | NONE

See Also

ACHOICE(), @...MENU

Description

The SET BORDER command changes the bordering of menus and @... commands from the default setting. The SINGLE option will draw single lines, whereas the DOUBLE option will draw double lines around menus and boxes. The PANEL option will cause the box to be drawn in reverse video. The NONE option disables bordering altogether.

An option must be specified with the SET BORDER command. The SET BORDER TO NONE command can be used to disable the default bordering of menus such as in the ACHOICE() function.

Example

```
// menu framed with double line
set border to double
menu1 = achoice(4,5,15,20,files)
```

Products

Recital Mirage Server, Recital Terminal Developer

SET CACHELOAD

Class

Performance and Optimization

Purpose

Map table into memory

Syntax

SET CACHELOAD ON | OFF | (<expL>)

See Also

SET DCACHE, SET ICACHE, SET PCACHE

Description

The SET CACHELOAD command can be used in association with the SET DCACHE command. It loads the table and index cache into memory when the table is used exclusively. Although this can be a lengthy process, further access to the table is accelerated dramatically. OpenVMS users please note that to obtain the full benefit of SET CACHELOAD, your working set should be set to a suitably large value. By default, CACHELOAD is OFF.

Example

```
set exclusive on
set dcache to 500
set dcache on
set cacheload on
use patrons index events
set dcache off
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CAPTURE

Class

Keyboard Events

Purpose

Capture all keyboard input in a keyboard macro file

Syntax

SET CAPTURE TO [<.kbm filename> | (<expC>)]

SET CAPTURE ON | OFF | (<expL>)

See Also

REPLAY, SET STATUS

Description

The SET CAPTURE TO <.kbm filename> command initiates the capturing of all keys pressed at the keyboard to be stored in a keyboard macro file. The filename can be substituted with an <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.kbm' is used. Everything typed, whether at the '>' prompt or in a form or menu, is stored in a file. When keyboard capture is complete, issuing the SET CAPTURE TO command without specifying a <.kbm filename> closes the capture file. The REPLAY command then allows the keys to be played back.

The command SET CAPTURE ON or OFF enables and disables the recording of keystrokes into the capture file. By default, CAPTURE is OFF.

As an alternative, keyboard macros can be initiated from within forms and menus by pressing the key sequence [^OC], then replayed by pressing the key sequence [^OR]. This allows you to store standard form filling sequences and replay them when needed. The ^OC and ^OR key sequences are only recognized if SET STATUS is ON.

The output file created from the SET CAPTURE TO command is stored in a standard ASCII file that can be edited if required. All standard keystrokes are stored as the key pressed (i.e. an 'a' is stored as a and control characters like ^W are stored as [EXIT/SAVE]). This makes the file very easy to change manually if a mistake was made while creating the macro file.

Example

```
set capture to query
use patrons index events
menu query
set capture to
```

Products

Recital Terminal Developer

SET CARRY

Class

Screen Forms

Purpose

Enable or disable data carry over to the new record added to the table

Syntax

SET CARRY ON | OFF | (<expL>)

See Also

APPEND, INSERT, SET FORMAT

Description

If SET CARRY is ON then the contents of the previously accessed record are carried over to the next one during the APPEND and INSERT commands. If SET CARRY is ON, the contents of the current record will be included in the new record created with the APPEND BLANK, APPEND BLANK <expN>, INSERT BLANK or INSERT BEFORE commands. Alternatively, when the APPEND or INSERT screens are active, pressing the [CARRY MODE] key toggles CARRY ON and OFF from within the form. By default, CARRY is OFF.

Values for specified fields can be pre-defined in the Applications Data Dictionary with the DEFAULT option. However, these default values will be overwritten if SET CARRY is ON.

Example

```
set carry on
use patrons
append
```

Products

Recital Mirage Server, Recital Terminal Developer

SET CATALOG

Class

Information Center

Purpose

Open or create a catalog for the Recital Information Center

Syntax

SET CATALOG TO <.cat filename> | (<expC>)

SET CATALOG ON | OFF | (<expL>)

See Also

DESIGN, INFO, SET DESIGN, SET TITLE

Description

The SET CATALOG TO command opens an existing catalog or creates a new catalog to display when the Recital Information Center initiates. The character expression (<expC>) is the name of an existing catalog, or the name of a catalog that you wish to create. A catalog is an object used to group and access files that pertain to a single application. Catalogs allow you to define your work environment by adding only those files that are needed, and excluding unnecessary files. The Recital Information Center displays catalogs by listing filenames in panels of the following categories: Data, Text, Form, Report, Label, and Program. One file can exist in more than one catalog. Each catalog entry may be customized with a user-defined title and description that are specific to that catalog. Removing a file from a catalog does not delete it from its directory.

Each catalog is actually a table file composed of nine fields. As files are added to the catalog through the Recital Information Center, the table file is automatically updated. The catalog table can be used just as you would USE any Recital table. A catalog table allows you to log detailed information regarding each file in the catalog, and generate system documentation.

SET CATALOG ON causes all newly created files to be automatically added to the currently open catalog. The SET CATALOG OFF command closes the currently open catalog, and disables automatic file adding. Catalogs may be opened from within the Information Center. When SET CATALOG is OFF, created files must be added with the <Add file to catalog...> option in the Options menu of the Information Center menu bar.

Example

set catalog to sales

set catalog on

design

Products

Recital Terminal Developer

SET CENTURY

Class

Date and Time Data

Purpose

Enable century in dates

Syntax

SET CENTURY ON | OFF | (<expL>)

See Also

CDOW(), CMONTH(), CTOD(), DATE(), DATETIME(), DAY(), DAYS(), DMY(), DOW(), DTOC(), DTOM(), DTOS(), DTOV(), EPOCH(), GOMONTH(), MDY(), MONTH(), MTOD(), QUARTER(), STOD(), VTOD(), YEAR(), SET DATE, SET EPOCH, SET MARK, SET SECONDS

Description

If SET CENTURY is ON, then dates are displayed, and can be input with the century prefix specified. If CENTURY is OFF then the year part of dates is only two digits, and the 20th century is assumed. By default, CENTURY is ON.

Example

```
set century on
use patrons index dates
list all for date = ctod("01/01/2003")
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CLASSLIBRARY

Class

Environment

Purpose

Open one or more class library files

Syntax

SET CLASSLIBRARY TO [<.cls filename> | (<expC>) [ADDITIVE]]

See Also

CLASS

Description

The SET CLASSLIBRARY TO <.cls filename> command opens the specified class library file, scans the contents of it, and records the names and positions of the classes defined within it. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. You can place as many classes as required in a class library file.

The settings of SET COMPILE and SET DEVELOPMENT (and Operating System permissions and disk space) will determine whether class library files are automatically compiled when the SET CLASSLIBRARY TO command is issued. Compiled class libraries have a '.clo' file extension.

If the optional ADDITIVE keyword is specified then any class libraries that are already open are left open and the new class library is added. Without the ADDITIVE keyword, existing class libraries will be closed when a new library is opened with the SET CLASSLIBRARY command. You can open up to 20 class library files at any one time. The SET CLASSLIBRARY TO command, without any <.cls filename> specified, closes all active class library files. A closed library file discards any knowledge of where the classes within reside.

Classes are declared by the CLASS keyword followed by the class name. The ENDCCLASS statement denotes the end of a class definition. No syntax checking is carried out while the class library file is being scanned. The syntax checking is only carried out when the class is instantiated. The active classes can be listed with the LIST | DISPLAY CLASSES command.

Example

```
// myclasslib.cls
class myclass
...
endclass

class myotherclass
...
endclass
// End of myclasslib.cls

set classlibrary to myclasslib
myobject = new myclass()
myobject2 = new myotherclass()
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CLEAR

Class

Recital Terminal Developer Environment

Purpose

Control whether or not the screen should be cleared on exit from Recital Terminal

Syntax

SET CLEAR ON | OFF | (<expL>)

See Also

QUIT

Description

The command SET CLEAR controls whether the screen is cleared when exiting. If CLEAR is set OFF then the screen will not be cleared on exit. If CLEAR is set ON, then the screen is cleared as you exit from the system. The default setting for CLEAR is ON.

Example

```
// Quit if ABANDON was pressed
if lastkey()=ctrl('g')
    set clear off
    quit
endif
```

Products

Recital Terminal Developer

SET CLIPPER

Class

Xbase Compatibility

Purpose

Support added Clipper compatibility

Syntax

SET CLIPPER ON | OFF | (<expL>)

See Also

CREATE, GOTO, PARAMETERS, PRIVATE, PUBLIC, SKIP, ASTORE(), EDITFIELD(), EOF(), FILE(), FILETYPE(), FOUND(), INDEXEXT(), MEMOEDIT(), SECONDS(), SECS(), VERSION(), SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET CLIPPER commands toggles between use of Clipper behavior and Recital behavior. There are certain instances where Clipper programs will behave slightly differently under the Recital environment. This SET command has been implemented to compensate for these differences.

When SET CLIPPER is ON the following will be in effect:

- The ASTORE() function will not trim the strings that are stored in the array passed to ASTORE().
- The FILE() function will not concatenate '.dbf' to the parameter, if the parameter passed does not have a file extension.
- The SECONDS() function will return the same value as the Recital function SECS().
- If less parameters are passed to a procedure or a function then are declared with the PARAMETER statement, the extra variables are set to type UNDEFINED instead of LOGICAL.
- If the record number qualifier of the GOTO command is less than zero or greater than the number of records, an error message is not generated and EOF() is set to TRUE and FOUND() is set to FALSE.
- In a program the command CREATE <filename> will create an empty database with the following structure.

Field	Type	Length	Description
FIELD_NAME	Character	32	field name
FIELD_TYPE	Character	1	data type of field
FIELD_LEN	Numeric	3	width of field
FIELD_DEC	Numeric	3	# of decimal places
FIELD_DESC	Character	25	field description

- Memory variables created with a PUBLIC or a PRIVATE statement are created with a type of UNDEFINED.
- The SKIP 0 command flushes locked records to disk if modifications have been made.
NOTE: SKIP 0 will not work this way in interactive mode.
- The MEMOEDIT() function provides full parameter syntax compatibility with Clipper
The VERSION() function will have a return value of 'Clipper Summer'87'.

Example

set clipper on
create strutable
set clipper off

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CLIPPER5

Class

Xbase Compatibility

Purpose

Support added Clipper 5 compatibility

Syntax

SET CLIPPER5 ON | OFF | (<expL>)

See Also

CREATE, GOTO, PARAMETERS, PRIVATE, PUBLIC, SKIP, ASTORE(), EDITFIELD(), EOF(), FILE(), FILETYPE(), FOUND(), INDEXEXT(), MEMOEDIT(), SECONDS(), SECS(), VERSION(), SET CLIPPER, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET CLIPPER commands toggles between use of Clipper 5 behavior and Recital behavior. This SET command has been implemented to compensate for the differences between the Recital and Clipper 5 4GLs.

When SET CLIPPER5 is ON the following will be in effect:

The ASTORE() function will not trim the strings that are stored in the array passed to ASTORE().

The FILE() function will not concatenate '.dbf' to the parameter, if the parameter passed does not have a file extension.

The SECONDS() function will return the same value as the Recital function SECS().

If fewer parameters are passed to a procedure or a function than are declared with the PARAMETER statement, the extra variables are set to type UNDEFINED instead of LOGICAL.

If the record number qualifier of the GOTO command is less than zero or greater than the number of records, an error message is not generated and EOF() is set to TRUE and FOUND() is set to FALSE.

In a program, the command CREATE <filename> will create an empty database with the following structure.

Field	Type	Length	Description
FIELD_NAME	Character	32	field name
FIELD_TYPE	Character	1	data type of field
FIELD_LEN	Numeric	3	width of field
FIELD_DEC	Numeric	3	# of decimal places
FIELD_DESC	Character	25	field description

Memory variables created with a PUBLIC or a PRIVATE statement are created with a type of UNDEFINED.

The SKIP 0 command flushes locked records to disk if modifications have been made.

NOTE: SKIP 0 will not work this way in interactive mode.

The MEMOEDIT() function provides full parameter syntax compatibility with Clipper.
The VERSION() function will have a return value of 'Clipper 5'.

Curly braces '{}' are interpreted as array size definition separators, not date delimiters.

Example

```
set clipper5 on  
create strutable  
set clipper5 off
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CLOCK

Class

Recital Terminal Developer Environment

Purpose

Specify details of screen display of time

Syntax

SET CLOCK ON | OFF | (<expL>)

SET CLOCK TO DEFAULT | <expN1>, <expN2>

See Also

SET CLOCKDISPLAY, SET CLOCKRATE

Description

The time on the screen can be updated at specified intervals. If SET CLOCK is ON, then the time is updated on the screen during terminal 'read' states in forms, menus, etc. By default, CLOCK is ON.

By default, the time is assumed to be displayed at row 0, column 69, but it may be repositioned with the command SET CLOCK TO row <expN1>, column <expN2>. SET CLOCK TO DEFAULT will reposition the clock display to the default position 0,69. To specify the interval at which the time display is updated, see the SET CLOCKRATE TO command.

Example

```
set clock on
set clock to 2,65
```

Products

Recital Terminal Developer

SET CLOCKDISPLAY

Class

Recital Terminal Developer Environment

Purpose

Turn clock display on or off

Syntax

SET CLOCKDISPLAY <ON | OFF>

See Also

SET STATUS, SET MESSAGE, SET HOURS

Description

The SET CLOCKDISPLAY command allows the lock to remain on for timers, but the clock is not displayed on the screen when CLOCKDISPLAY is set OFF. This is particularly useful for ON TIMEOUT and mail processing when no display of the clock is desired. SET CLOCKDISPLAY is ON by default.

Example

```
set clockdisplay off
```

Products

Recital Terminal Developer

SET CLOCKRATE

Class

Recital Terminal Developer Environment

Purpose

Set update intervals for screen display of time

Syntax

SET CLOCKRATE TO <expN>

See Also

SET CLOCK

Description

The time on the screen can be updated at specified intervals. If SET CLOCK is ON, then the time is updated on the screen during terminal READ states in forms, menus, etc. The SET CLOCKRATE TO <expN> command allows you to specify the interval at which the time display is updated in seconds. By default, CLOCKRATE is 1.

Example

set clock on
set clockrate to 5

Products

Recital Terminal Developer

SET COLOR

Class

Screen Environment

Purpose

Specify screen colors

Syntax

SET COLOR TO [[<standard>] [, [<enhanced>] [, [<border>]]]] | (<expC>)

SET COLOR ON | OFF | (<expL>)

See Also

CLEAR, @...FILL, @...GET, @...SAY, @...TO, SAVE COLOR, RESTORE COLOR, ISCOLOR(), SETCOLOR()

Description

The SET COLOR command controls the color and display attributes of the screen. You can define the color settings of the screen or individual display areas on the screen. The color settings are changed by specifying the letter codes for each color.

Color	Attribute Letter
BLACK	N or blank
BLUE	B
CYAN	BG
BLANK	X
GREY	N+
RED	R
MAGENTA	RG
BROWN	GR
YELLOW	GR+
WHITE	W

Each color code can be specified as a pair, separated with a '/', which will change the foreground/background color.

The SET COLOR TO (<expC>) is used to assign color settings to memory variables. In this way users may select color settings.

The SET COLOR TO <standard> command will change the standard color display areas on the screen. For machines that allow you to set different colors for standard and enhanced text areas, set the <standard> and <enhanced> colors in pairs as foreground/background colors. You can change one or more color attributes at the same time. The position of commas indicates the setting you want to change. For example, to set the enhanced area to white without affecting the other settings, you would type SET COLOR TO ,W. In specifying foreground/background pairs, you can omit the foreground or background color. When you do not specify a color, either before or after the '/' in the foreground/background pair, the color black is selected.

The SET COLOR TO command without qualifiers resets colors to default settings.

The SET COLOR ON | OFF command switches the color back and forth between monochrome and color.

Example

set color to GR+/B && set the color yellow on blue
background = "GR"
set color to (foreground + "/" + background)

Products

Recital Mirage Server, Recital Terminal Developer

SET COLOR OF

Class

Screen Environment

Purpose

Change the color of a predefined display area

Syntax

SET COLOR OF <display area> TO [<color>]

SET COLOR OF SCHEME <expN> TO [<color> | SCHEME <expN>]

SET COLOR SET TO [<expN>]

See Also

CLEAR, @...FILL, @...GET, @...SAY, @...TO, SAVE COLOR, RESTORE COLOR, ISCOLOR(), SETCOLOR(), SET COLOR

Description

The SET COLOR OF <display area> TO <color> command controls predefined screen area groupings so that you can change color attributes of individual screen areas independent of one another. The following display areas are predefined.

Display Area	Description
ALERT	All error message alert boxes
BOX	All boxes
BROWSE	BROWSE work surface
BROWSEFRAME	BROWSE frames
DIALOG	Dialog boxes
[DIALOG]FRAME	Frames around dialog boxes
FIELDS	Normal display area like <standard>
GETFOCUS	Current get field
HELP	Message line
HIGHLIGHT	Highlighted menu options
INFORMATION	Status bar, clock, etc
MENUS	Pulldown, pop-up menus
MESSAGES	System message bar
NORMAL	Normal screen output
SHADOWS	Bottom and right edge of dialog boxes
TITLES	Dialog box titles

The SET COLOR OF SCHEME and SET COLOR SET commands have been added for syntax compatibility with FoxPro.

Example

```
set color of menu to W      && set menus to white
foreground = "N"           &&use memory variables
background = "GR"
set color to (foreground + "/" + background)
```

Products

Recital Mirage Server, Recital Terminal Developer

SET COMMAND

Class

Recital Terminal Developer Environment

Purpose

Specify command identity in the status bar

Syntax

SET COMMAND TO <expC>

See Also

SET STATUS

Description

The SET COMMAND TO command updates the first box of the status bar with the specified character string <expC> up to 15 characters in length. If SET COMMAND TO is used on its own, the status bar is updated with the default text.

Example

set command to “myprog”

Products

Recital Terminal Developer

SET COMMANDWINDOW

Class

Screen Windows

Purpose

Enable or disable command windows

Syntax

SET COMMANDWINDOW ON | OFF | (<expL>)

See Also

ACTIVATE SCREEN, ACTIVATE WINDOW, CLEAR WINDOWS, DEACTIVATE WINDOW, DEFINE WINDOW, HIDE WINDOW, MOVE WINDOW, MODIFY MEMO, RELEASE WINDOWS, RESIZE WINDOW, RESTORE WINDOW, SAVE WINDOW, SHOW WINDOW, SET ERRORWINDOW, SET STATUS, SET TRACEWINDOW, SET WINDOW OF EDIT, SET WINDOW OF MEMO, WROWS(), WCOLS(), WEXIST(), WVISIBLE(), WONTOP(), WOUTPUT()

Description

The SET COMMANDWINDOW command is used to enable or disable the use of command windows. A window is an area of the screen designated for output and input. There is no limit to the number of windows you may define. Windows are defined with the DEFINE WINDOW command, and are activated with the ACTIVATE WINDOW command. A command window is a window that automatically activates itself and displays the interactive command prompt. When active, command windows allow the input of interactive commands. When SET COMMANDWINDOW is ON, a pre-defined command window automatically activates when control returns to the interactive command prompt. When SET COMMANDWINDOW is OFF, the command window does display, and the interactive prompt appears in full screen mode rather than in a window.

Example

```
define window command;  
  from 17,45 to 23,79;  
  color w/r;  
  panel;  
  shadow;  
  float;  
  grow;  
  command  
set commandwindow on
```

Products

Recital Terminal Developer

SET COMPATIBLE

Class

Xbase Compatibility

Purpose

Enhance Recital's compatibility with other Xbase products

Syntax

SET COMPATIBLE [[TO] ON | OFF]

SET COMPATIBLE TO DBASE3 | DBASE4 | DB4 | FOXPRO | FOXPLUS | FOXBASE | CLIPPER |
CLIPPER5 | RECITAL | VFP

See Also

APPEND, CHANGE, EDIT, EDITFIELD(), FILETYPE(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXMEM, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET COMPATIBLE command is used to enhance compatibility with other Xbase products. Certain functions and commands behave differently depending on which keyword is used with the SET COMPATIBLE command.

In Recital Terminal Developer, if SET COMPATIBLE is issued with no ON, OFF or TO clause, the Xbase Compatibility Settings dialog is displayed. This allows the system-wide Language and Database Compatibility settings to be selected. These correspond to the SET COMPATIBLE and SET FILETYPE settings and once saved are written to the *compat.db* file, which will be run every time Recital Terminal Developer is started. The *compat.db* file resides in the Recital Terminal Developer home directory. The following choices are available:

- Recital
- Visual FoxPro
- FoxPro
- FoxBase
- dBASE IV
- dBase 3
- Clipper 5
- Clipper '87

The Xbase Compatibility Settings dialog is also displayed the first time you start Recital Terminal Developer after a new install or at any subsequent startup when the *compat.db* file does not exist in the Recital Terminal Developer home directory.

The keywords ON, DBASE4, and DB4 perform the same function, that is, provide compatibility with dBase IV.

When SET COMPATIBLE is set to any mode except RECITAL, the following SET commands are turned on. SET PROMPT is also set to '.'.

- SET PCEXACT
- SET PCUNIQUE
- SET PCFILTER
- SET PCLOCKING
- SET PCKEYS

In any compatibility mode other than RECITAL, the AGAIN clause of the USE command allows the same table to be reopened in another workarea in the same session.

Language	Memo	Index	Production
DBASE3	dbt	ndx	mdx
DBASE4	dbt	ndx	mdx
CLIPPER	dbt	ntx	
FOXBASE	dbt	idx	cdx
FOXPRO	fpt	idx	cdx
VFP	fpt	idx	cdx

When COMPATIBLE is set to FOXPRO | VFP, the following changes apply:

ADIR()

ADIR() is compatible with FoxPro behavior.

DBF()

If SET FULLPATH is ON the full path to the database table is returned, if SET FILECASE is OFF then the return value is converted to upper case (Windows only).

FMT()

If SET FULLPATH is ON the full path to the format file is returned, if SET FILECASE is OFF then the return value is converted to upper case (Windows only).

MDX()

If SET FULLPATH is ON the full path to the index file is returned, if SET FILECASE is OFF then the return value is converted to upper case (Windows only).

NDX()

If SET FULLPATH is ON the full path to the index file is returned, if SET FILECASE is OFF then the return value is converted to upper case (Windows only).

When COMPATIBLE is set to FOXBASE | FOXPLUS | FOXPRO | VFP, the following changes apply:

File specifications

File specifications can also be character expressions, e.g. APPEND FROM DBF(2). In other compatibility modes, the character expression must be enclosed in parentheses, e.g. APPEND FROM (DBF(2)).

==

The == operator always operates as an 'exact match' operator and does not allow wildcard pattern matching.

AFIELDS()

If SET COMPATIBLE is set to FOXPRO | FOXPLUS | FOXBASE | VFP the array need not be pre-defined, it will be created automatically.

COPY TO ARRAY

If SET COMPATIBLE is set to FOXPRO | FOXPLUS | FOXBASE | VFP the array need not be pre-defined, it will be created automatically.

DTOS()

The DTOS() function, when used on an empty date, will return a string of 8 spaces rather than a string of 8 zeroes.

SET FULLPATH

SET FULLPATH is set ON by default.

When compatibility is set to DBASE4 or to FOXBASE | FOXPLUS | FOXPRO | VFP the following changes apply:

ARRAY

The word ARRAY is recognized as an optional keyword to the LOCAL, PUBLIC, PRIVATE and DECLARE commands when compatibility is set to DBASE4 or FOXPRO. Normally, Recital allows the word ARRAY to be used as a memory variable name.

BROWSE

When BROWSE is activated in a window the BROWSE menu will not be shown until the [MENUBAR] key is pressed. Normally the BROWSE menu is displayed when BROWSE is activated.

INDEXORDER() | ORDER()

The INDEXORDER() and ORDER() functions will return the name of the master .ndx or master .dbx tag instead of the ordinal number of the index in the index list.

MENU()

The MENU() function will return the name of the activated menu instead of the ordinal number of the menu item chosen.

PROGRAM() | PROCNAME()

The PROGRAM() and PROCNAME() functions return a string in upper case instead of a string in lower case.

SELECT([<expN>])

If <expN> is 0, the SELECT() function will return the currently selected workarea. If <expN> is 1, the SELECT() function will return the highest available empty workarea.

SET()

The SET() function will return a string instead of a logical value. In many cases, when SET COMPATIBLE is set to DB4, SET() will return ON or OFF. In other cases SET() will return a setting, for example, SET("BORDER") might return DOUBLE.

SYS(5)

The SYS(5) function will return 'C:'

When SET COMPATIBLE is set to CLIPPER the following commands and functions are affected.

ASTORE()

The ASTORE() function will not trim the strings that are stored in the array passed to ASTORE().

CREATE

In a program the command CREATE <filename> will create an empty table with the following structure.

Name	Type	Width	Description
FIELD_NAME	Character	32	Field name
FIELD_TYPE	Character	1	Field data type
FIELD_LEN	Numeric	3	Width of fields
FIELD_DEC	Numeric	3	Number of decimals

FILE()

The FILE() function will not concatenate '.dbf' to the parameter, if the parameter passed does not have a file extension.

GOTO

If the record number qualifier of the GOTO command is less than zero or greater than the number of records, an error message is not generated and EOF() is set to TRUE and FOUND() is set to FALSE.

MEMOEDIT()

The parameters that are passed to MEMOEDIT() are compatible with CLIPPER.

PARAMETER

If fewer parameters are passed to a procedure or a function than are declared with the PARAMETER statement, the extra variables are set to type UNDEFINED instead of LOGICAL.

PUBLIC | PRIVATE

Memory variables created with a PUBLIC or PRIVATE statement are created with a type of UNDEFINED. The TYPE() function will return "U" for undefined variables.

SAVE | RESTORE SCREEN

The SAVE and RESTORE SCREEN command without any qualifiers, will only SAVE and RESTORE one screen. Normally Recital allows 20 screens to be saved and restored.

SECONDS()

The SECONDS() function will return the same value as SECS() function.

SKIP 0

When a SKIP 0 command is executed, the record buffer is flushed to disk. This command is most useful when used in validation procedures that are called from forms and REPLACE commands are used within the validation procedure.

The VERSION() function will have a different value as outlined below:

Language	VERSION()
DBASE4 DB4	dBASE IV 2.0
CLIPPER	Clipper Summer '87
CLIPPER5	Clipper 5
FOXBASE FOXPLUS	FoxBASE+ 2.10
FOXPRO	FoxPro 2.6
VFP	Visual FoxPro 06.00 for Windows

{ }

When SET COMPATIBLE is set to CLIPPER5, curly braces '{ }' are interpreted as array size delimiters, not date delimiters.

When SET COMPATIBLE is set to VFP, the following changes apply:

ENDFUNC

The ENDFUNC command can be used to terminate a function. If no explicit return value is set, the return value will be .T. (true).

ENDPROC

The ENDPROC command can be used to terminate a procedure. If no explicit return value is set, the return value will be .T. (true).

FUNCTION | PROCEDURE

FUNCTION and PROCEDURE definitions in procedure libraries do not require a RETURN statement (or ENDFUNC | ENDPROC) as they will be terminated by the next FUNCTION | PROCEDURE definition. The FUNCTION command supports parameter declaration in the format FUNCTION function-name(parameter-list).

ID()

The ID() function returns a string in the following format, returning the machine id and the user id rather than just the user id:

MACHINEID # userid

Example

set compatible dbase4

?set("talk")

ON

set compatible recital

?set("talk")

.T.

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET COMPILE

Class

Environment

Purpose

Enable or disable automatic recompiling of modified source programs

Syntax

SET COMPILE ON | OFF | (<expL>)

See Also

COMPILE, SET DEVELOPMENT, DO, SET PSHARE

Description

The SET COMPILE command controls whether program files will be compiled as they are accessed, generating an object file that will then run. If SET COMPILE is ON, each time a program is run, it is compiled and an object file is generated. Files with a '.prg' extension have compiled object files with a '.dbo' extension, any other object files have the same extension as their source file, except that it ends with an 'o'. The object file, not the source file is then executed. When SET COMPILE is OFF, programs are run interpreted. Each line of the program is read and executed one after the other. By default, COMPILE is ON.

COMPILE should be set OFF when developing applications. COMPILE should be set ON once the application has been debugged and is running properly.

Example

```
set compile on
do main_prog
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CONFIRM

Class

Screen Environment

Purpose

Determine how the cursor moves to the next field when in a form

Syntax

SET CONFIRM ON | OFF | (<expL>)

See Also

SET BELL, @...SAY...GET, SET FORMAT

Description

If SET CONFIRM is ON, the [RETURN] key must be pressed to finish entering data into a field on a form. When CONFIRM is OFF, filling the field causes the cursor to move automatically to the start of the next field. If the field is the last active field on the screen, the cursor will not go to the first field of the next record. The [NEXT RECORD] key must be pressed to access the next record. By default, CONFIRM is OFF.

Example

```
set confirm on
use patrons
edit
```

Products

Recital Terminal Developer

SET CONSOLE

Class

Screen Environment

Purpose

Turn the screen display on or off

Syntax

SET CONSOLE ON | OFF | (<expL>)

See Also

@...GET NOECHO

Description

If SET CONSOLE is OFF, no output is displayed on the screen. Characters can be input so that, for example, reports may be printed on a slave printer attached to the terminal without the text appearing on the screen. Another usage is to read passwords without echo. By default, CONSOLE is ON.

Example

```
set console off
report form patrons to print
set console on
```

Products

Recital Terminal Developer

SET CURRENCY

Class

Numeric Data

Purpose

Define the currency symbol associated with the CURRENCY() function

Syntax

SET CURRENCY TO <expC>

See Also

CURRENCY()

Description

The SET CURRENCY TO command allows you to define the currency symbol associated with the CURRENCY() function or the '\$' picture symbol, as specified by the character expression <expC>. This expression has a length of one character. By default, CURRENCY is \$.

NOTE: The SET LANGUAGE command may be required to obtain the '£' character. See the SET LANGUAGE command for further details.

Example

```
set currency to '£'  
? currency(5000.00)  
£5000.00
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET CURSOR

Class

Keyboard Events

Purpose

Turn the cursor on or off

Syntax

SET CURSOR ON | OFF | (<expL>)

See Also

SET PROMPT

Description

The SET CURSOR command turns the cursor on or off. By default, CURSOR is ON.

Example

set cursor on

Products

Recital Terminal Developer

SET DATE

Class

Date and Time Data

Purpose

Determine input and output date formats

Syntax

SET DATE [TO] AMERICAN | ANSI | BRITISH | DMY | ITALIAN | JAPAN | FRENCH | GERMAN | MDY | USA | YMD

See Also

SET MARK, SET VAXTIME, DATE(), CTOD(), DTOC(), DTOS(), STOD(), DTOV(), VTOD(), DMY(), MDY()

Description

The SET DATE command specifies the input and output format for dates. The following summarizes the available date formats.

Date	Format
AMERICAN (or MDY)	mm/dd/yy
ANSI	yy.mm.dd
BRITISH or FRENCH or (DMY)	dd/mm/yy
GERMAN	dd.mm.yy
ITALIAN	dd-mm-yy
JAPAN or (YMD)	yy/mm/dd
USA	mm-dd-yy

If SET CENTURY is ON, then the years will be formatted with four digits, i.e. yyyy. The SET MARK command may be used to specify an alternate character to separate the parts of a date.

Example

```
set date german
use patrons index dates
seek ctod("01.01.98")
set century on
seek ctod("01.01.1998")
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DBTRAP

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET DBTRAP ON | OFF | (<expL>)

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

```
set dbtrap on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DCACHE

Class

Performance and Optimization

Purpose

Map table into memory

Syntax

SET DCACHE TO [<expN>] [OPTIMUM]

SET DCACHE ON | OFF | (<expL>)

See Also

SET CACHELOAD, SET GCACHE, SET ICACHE, SET PCACHE

Description

If SET DCACHE is ON when a table is used, then the table records will be 'cached' in memory, and the records which are not held in memory will be read in when they are needed.

The SET DCACHE TO <expN> command allows the number of records held in memory to be specified on a per table basis. If the optional OPTIMUM keyword is specified, the optimum dcache size will automatically be set. The table 'cache' area acts as a 'window' into the table. Each record referenced in the table is 'mapped' into the table cache using the simple formula:

$$\text{position} = \text{mod}(\text{<record number>}, \text{<cache size>})$$

Using this formula with a table cache of 500, record 1 maps into slot 1, record 501 also maps into slot 1.

When a slot is referenced which already contains a record, and that record has been modified since it was read from the table, then the table is updated with the modified record and the required record is read into the cache slot. When a table is closed, modified records held in memory are written into the table.

The SET DCACHE command has no effect if a table is shareable, unless SET GCACHE is set ON. When SET GCACHE is ON, SET DCACHE and SET ICACHE operate on a distributed basis. The SET CACHELOAD command can be used in association with the SET DCACHE command. This forcibly loads the table and index cache when the table is used exclusively. The SET ICACHE command can be used to 'cache' index keys in memory. By default, DCACHE is OFF.

Example

```
// Open up payroll system
set exclusive on
set cacheload on
set icache to 3000
set dcache to 1000
use payroll index pay_date, emp_code
// Cache the employee records in memory
set dcache on
use employees index name, emp_code
set dcache off
use wages index emp_code
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DEBUG

Class

Error Handling and Debugging

Purpose

Toggle sending of echoed commands to printer

Syntax

SET DEBUG ON | OFF | (<expL>)

See Also

SET ECHO, SET TALK, SET ALTERNATE, SET PRINT, SET PRINTER

Description

If SET DEBUG is ON and SET ECHO is ON, then each command that is read from a program file is displayed on the printer. This command is primarily used in program debugging. By default, DEBUG is OFF.

Example

```
set debug on
set echo on
do yourproc
```

Products

Recital Terminal Developer

SET DECIMALS

Class

Numeric Data

Purpose

Specify the number of decimal places to be used when displaying the results of numeric expressions

Syntax

SET DECIMALS TO <expN>

See Also

SET FIXED, ROUND()

Description

The SET DECIMALS command determines the minimum number of decimal places displayed in the result of numeric functions and calculations. For calculations that do not involve numeric functions, division, or multiplication, the number of decimal places displayed is the same as the number or variable with the most decimal places. For calculations involving multiplication, the number of decimal places displayed is the sum of the decimal places of the numbers being multiplied.

The SET DECIMALS command is normally used in conjunction with the SET FIXED command to specify a fixed number of decimal places for calculation results.

Example

```
? 285.129 * 4.6  
1311.5934  
set decimals to 3  
set fixed on  
? 285.129 * 4.6
```

```
1311.593
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DEFAULT

Class

Environment

Purpose

Specify default device name and directory on which files will be searched for and stored

Syntax

SET DEFAULT TO [<device name>] [<path-name>]

See Also

SET PATH, SET VIEW, DEFAULT(), SET DIRECTORY

Description

The SET DEFAULT command allows you to specify the device and directory where all files will be searched for and stored. The <device name> is the name of the device that you wish to be the default device. By default, files are searched for and stored in the current working directory. The SET DEFAULT TO command without a <path-name> returns you to the device and the directory that was originally active.

Example

```
set default to c:\usr\accounts\
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DELETED

Class

Fields and Records

Purpose

Determine whether or not records that are marked for deletion are hidden

Syntax

SET DELETED ON | OFF | (<expL>)

See Also

DELETE, RECALL, SET FILTER, DELETED()

Description

If SET DELETED is ON, then records that have been marked for deletion are ‘invisible’. Records can be marked for deletion with the DELETE command or the DELETE key, and recalled with the RECALL command or the DELETE key. Deleted records can be recalled until a PACK command is issued on the table. By default, DELETED is OFF.

Example

```
set deleted on
use patrons
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DELIMITERS

Class

Screen Forms

Purpose

Specify how field widths should be denoted in forms

Syntax

SET DELIMITERS TO <expC> | DEFAULT

SET DELIMITERS ON | OFF | (<expL>)

See Also

SET INTENSITY, SET UNDERLINE, @...SAY ...GET

Description

If SET DELIMITERS is ON, then fields displayed on forms will be delimited with ':' to identify the field width. The SET DELIMITERS TO command sets the delimiter characters to those specified. If only one character is specified, then it will be used for both the left and right of the fields. The SET DELIMITERS TO DEFAULT command resets the delimiters back to ':'. By default, DELIMITERS are OFF and the delimiter characters are ':'.

Example

```
set delimiters on
set delimiters to "["
use patrons index events
change
set catalog to project
info
```

Products

Recital Terminal Developer

SET DESCRIPTIONS

Class

Environment

Purpose

Enable or disable the display of field descriptions in full screen forms

Syntax

SET DESCRIPTIONS ON | OFF | (<expL>)

See Also

CREATE, CREATE SCREEN

Description

If SET DESCRIPTIONS is ON, then the field descriptions associated with a table field will be displayed as field prompts in forms. Default forms are generated for the APPEND, CHANGE, EDIT, INSERT and QUERY commands if none have been specified with the SET FORMAT TO command. The field descriptions will also be displayed in user-defined forms and field menus resulting from the MENU QUERY and MENU FIELDS commands.

Field descriptions can be toggled ON and OFF from within the form by selecting the *Descriptions* option in the menu bar. Field descriptions are specified when a table is created using the CREATE command. User defined forms are created with the Screen Painter, which is activated by the CREATE SCREEN command. When SET DESCRIPTIONS is ON, field descriptions rather than field names are loaded on to the Screen Painter work surface. If SET DESCRIPTIONS is ON, and SET HEADING is set to ON, the field descriptions will display as the column titles. By default, DESCRIPTIONS are OFF.

Example

```
use patrons index events
set descriptions on
change all for event = "BALLET"
```

Products

Recital Terminal Developer

SET DESIGN

Class

Information Center

Purpose

Set the mode, design or info, for the Recital Information Center

Syntax

SET DESIGN ON | OFF | (<expL>)

See Also

SET CATALOG, DESIGN, INFO

Description

The SET DESIGN command determines whether the Recital Information Center will initiate in Design Mode or not. The Recital Information Center is initiated with the INFO or DESIGN commands.

The Recital Information Center is a powerful work surface, from which any type of data may be accessed, viewed, modified and organized. With the Information Center you can organize files into catalogs which represent a single application, an application interface or a development project. Depending on your data requirements, the Information Center may serve as an application interface, or as a development tool.

Design Mode allows access to CREATE and MODIFY work surfaces as well as many other design tools.

Design Mode may also be enabled with the DESIGN command. When in Design Mode, the Information Center displays a [DESIGN] key that is used to access development work surfaces. Design Mode also displays program files with an extension of '.prg' or '.app'. When Design Mode is disabled, only program files with an extension of .app display, and files with an extension of .prg display only in catalogs to which they have been added.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, DESIGN is set ON. If a value of .F. is returned, DESIGN is set OFF. Note that <expL> should be enclosed in round brackets.

Example

set design on

Products

Recital Terminal Developer

SET DEVELOPMENT

Class

Environment

Purpose

Determine which files need to be compiled

Syntax

SET DEVELOPMENT ON | OFF | (<expL>)

See Also

COMPILE, SET COMPILE, DO, SET PSHARE

Description

When DEVELOPMENT is set ON, the modification date and time of the program source file is compared with the modification and time of the compiled object file (.dbo) when a program is executed. This prevents an outdated object file from executing. If the .dbo file is older than its source program file, the source file is automatically recompiled. If SET DEVELOPMENT is OFF then existing '.dbo' files will be executed even if there is a corresponding source file that is newer. For source files that have no corresponding '.dbo' file, one will be created. By default, DEVELOPMENT is ON.

Example

```
set compile on
set development on
do prototype
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DEVICE

Class

Input/Output

Purpose

Specify whether @...SAY commands are sent to the printer

Syntax

SET DEVICE TO SCREEN | PRINT | FILE <filename>

See Also

SET PRINT, SET PRINTER, PRINT, REPORT, TREPORT, @...SAY

Description

The SET DEVICE TO PRINT command sends the output from @...SAY commands to the printer. After this command has been issued, the PROW() and PCOL() functions return the current row and column position of the print head on the printer, and the ROW() and COL() functions return undetermined values. When DEVICE is set to PRINT, @...GET commands are ignored.

The SET DEVICE TO SCREEN command sends the output from @...SAY commands to the screen. After this command has been issued, the ROW() and COL() functions return the current row and column position of the cursor on the screen, and the PROW() and PCOL() functions return undetermined values.

The SET DEVICE TO FILE command directs @...SAY commands to a file specified by <filename>.

By default, SET DEVICE is set to SCREEN, which is the normal mode of operation.

Example

```
set device to print
eject
@0,0 say center("SALES REPORT ", 132)
@1,0 say center("-----", 132)
set device to screen
```

```
// Another example
// Send @...SAYs to a slave printer
set screenmap off
set console off
set device to print
set print on
// Issue @...SAYs here...
set print off
set device to screen
set console on
set screenmap on
```

Products

Recital Terminal Developer

SET DIALOG

Class

Screen Environment

Purpose

Enable or disable dialog style messages

Syntax

SET DIALOG ON | OFF | (<expL>)

See Also

DIALOG MESSAGE, DIALOG QUERY, DIALOG FILES LIKE, DIALOG FIELDS, DIALOG SCOPE

Description

The SET DIALOG OFF command disables all dialog and alert boxes. Dialog boxes display when the user presses the [ABANDON] key. Alert boxes display when an error occurs. Error messages display in the message line rather than in a pop-up alert box when SET DIALOG is OFF. SET DIALOG OFF is particularly useful in foreign language applications. The default for SET DIALOG is ON.

Example

```
set dialog off
```

Products

Recital Mirage Server, Recital Terminal Developer

SET DICTIONARY

Class

Table Organization

Purpose

Change or deactivate the active dictionary

Syntax

```
SET DICTIONARY ON | OFF | (<expL>)  
SET DICTIONARY TO [<.dbd filename> | (<expC>)]
```

See Also

CREATE, MODIFY STRUCTURE

Description

The SET DICTIONARY ON | OFF command enables or disables the use of Recital's Data Dictionary (.dbd) files. By default SET DICTIONARY is ON. When SET DICTIONARY is OFF, and a table is opened that has an associated data dictionary, the data dictionary is ignored. To disable a dictionary file after the table has opened, use the SET DICTIONARY TO command.

The SET DICTIONARY TO <.dbd filename> command can be used to change the current dictionary. The filename can be substituted with an <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.dbd' is used. This command will change the active dictionary to the specified <.dbd filename>. The dictionary must have been created from the current table.

The dictionary file is created from the CREATE or MODIFY STRUCTURE work surface. It is created with the same basename as the table, but with a '.dbd' extension. To create another dictionary for the same table, the current dictionary should be copied to a new name first.

The SET DICTIONARY TO command will deactivate the current dictionary.

NOTE: SET DICTIONARY TO will not deactivate triggers or field and table protection.

Example

```
set dictionary off  
use customer.rdb  
copy to cust2  
use cust2  
set dictionary to customer
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DIRECTORY

Class

Environment

Purpose

Specify default device name and directory in which files will be searched for and stored

Syntax

SET DIRECTORY TO [<device name>] [<path-name>]

See Also

SET PATH, SET VIEW, DEFAULT(), SET DEFAULT

Description

The SET DIRECTORY command allows you to specify the device name and directory where all files will be searched for and stored. The device name specification is optional and if present, is ignored on non OpenVMS platforms. By default, files are searched for and stored in the current directory. The SET DIRECTORY command without a <path name> returns you to the original device and directory.

Example

set default to C:\accounts

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET DISPLAY

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET DISPLAY TO MONO | COLOR | EGA25 | EGA43 | MONO43

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

set display to mono

Products

Recital Mirage Server, Recital Terminal Developer

SET DOESCAPE

Class

Keyboard Events

Purpose

Determine what happens when the interrupt key is pressed

Syntax

SET DOESCAPE ON | OFF | (<expL>)

See Also

SET ESCAPE, ON ESCAPE, ON KEY, INKEY()

Description

If SET DOESCAPE is ON, and ESCAPE is ON, and a command is executing in a program file, when the interrupt key is pressed, the command is canceled and execution of the program continues with the next command. The DOESCAPE command differs from the ESCAPE command in that interruption of a program only cancels the current command being executed. If ESCAPE is OFF, SET DOESCAPE has no effect. By default, DOESCAPE is OFF.

Example

```
//program longsort
set doescape on
set escape on
sort on name /a, event /d, date /a
return
```

```
use patrons
```

```
do longsort
```

Products

Recital Terminal Developer

SET DOHISTORY

Class

Error Handling and Debugging

Purpose

Determine whether commands from program files are recorded in history

Syntax

SET DOHISTORY ON | OFF | (<expL>)

See Also

SET HISTORY, SUSPEND, RESUME, SET STEP, SET ECHO, SET DEBUG, SET PRINT

Description

If SET DOHISTORY is ON, and HISTORY is ON, then commands executed in program files are recorded in the command history. This command is primarily used during program debugging. The commands executed in the program, while DOHISTORY is ON, can be listed with the LIST HISTORY command. If used in conjunction with the SET HISTORY TO FILE <.his filename> the program history can be stored in a text file. By default, DOHISTORY is OFF.

Example

```
set history on
set dohistory on
do testprg
list history
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ECHO

Class

Error Handling and Debugging

Purpose

Toggle display of commands from programs as they are being executed

Syntax

SET ECHO ON | OFF | (<expL>)

Description

If SET ECHO is ON, then commands from program files are echoed to the screen | printer as they are executed. This command is primarily used during program debugging. By default, ECHO is OFF.

See Also

SET STEP, SET DEBUG, SET HISTORY, SET DOHISTORY, SET TALK, SUSPEND, RESUME

Example

```
set echo on
set step on
do testprg
```

Products

Recital Terminal Developer

SET EDITFIELD

Class

Screen Forms

Purpose

Change the field editing operation.

Syntax

SET EDITFIELD ON | OFF | (<expL>)

See Also

APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), READEXIT(), READINSERT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

A synonym of the EDITFIELD() function, this command enables or disables dBASE compatible use of the [CURSOR RIGHT] key for editing of textual fields on forms. When EDITFIELD is ON, characters initially typed over a field overwrite only those characters shown, and on pressing the [RETURN] key the remainder of the field is included as user input. If EDITFIELD is set OFF, only what is typed into the field is saved. The default setting for EDITFIELD is OFF.

Example

```
set editfield on
use accounts
set form to customers
```

Products

Recital Terminal Developer

SET EMACROS

Class

Environment

Purpose

Enable or disable & macro substitution

Syntax

SET EMACROS ON | OFF | (<expL>)

See Also

&, TEXT, SET MACROS

Description

The SET EMACROS command is synonymous with SET MACROS. If SET EMACROS is ON, & macros are substituted when command lines are read. If SET EMACROS is OFF, then no macro substitution is performed.

SET EMACROS OFF can be used to marginally accelerate program execution and also disable macro substitution in TEXT...ENDTEXT blocks.

By default, SET EMACROS is ON.

Example

```
set emacros off
text
```

The & command is used to perform macro substitution of the contents of memory variables.

```
endtext
set emacros on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ENCRYPTION

Class

DES3 Encryption

Purpose

Specify the default encryption key to be used when an encrypted table is accessed

Syntax

```
SET ENCRYPTION TO [<expC>]  
SET ENCRYPTION ON | OFF | (<expL>)
```

See Also

APPEND FROM, COPY FILE, COPY STRUCTURE, COPY TO, DECRYPT, DIR, ENCRYPT, USE

Description

If a database table is encrypted, the correct three-part encryption key must be specified before the table's data or structure can be accessed. The SET ENCRYPTION TO set command can be used to specify a default encryption key to be used whenever an encrypted table is accessed without the key being specified. The <expC> is the three part comma-separated key.

If the command to access the table includes the key, either by appending it to the table filename specification or using an explicit clause, this will take precedence over the key defined by SET ENCRYPTION TO.

Issuing SET ENCRYPTION TO without the <expC> causes any previous setting to be cleared. The key must then be specified for each individual encrypted table.

The default key defined by SET ENCRYPTION TO <expC> is only active when SET ENCRYPTION is ON. SET ENCRYPTION OFF can be used to temporarily disable the default key. The SET ENCRYPTION ON | OFF setting does not change the default key itself. SET ENCRYPTION is ON by default.

Example

```
encrypt table1 key "key_1,key_2,key_3"  
encrypt table2 key "key_2,key_3,key_4"  
set encryption to "key_1,key_2,key_3"  
use table1  
use table2<key_2,key_3,key_4>  
set encryption to  
use table1 encryption "key_1,key_2,key_3"  
use table2<key_2,key_3,key_4>
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET EPOCH

Class

Date and Time Data

Purpose

Specify a one hundred-year epoch to allow for two digit years.

Syntax

SET EPOCH TO <expN>

See Also

SET CENTURY, SET DATE, DATE(), CTOD(), DTOC()

Description

The SET EPOCH TO <expN> command allows for the specification of the starting year of a one hundred-year period. The one hundred-year period can span two consecutive centuries. Two-digit year dates (SET CENTURY OFF) can be used and still assigned the correct century provided that they fall within this one-hundred year period. For example, SET EPOCH TO 1995 would specify a one hundred-year period from 1995 to 2094. Any two-digit years less than 95 would be 21st century, any two-digit years of 95 or over would be 20th century.

10/10/97 = October 10th 1997

10/10/27 = October 10th 2027

Example

set century off

set epoch to 1995

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ERRORVERSION

Class

Error Handling and Debugging

Purpose

Cause numbered error.mem files to be created when an error occurs

Syntax

SET ERRORVERSION ON | OFF | (<expL>)

See Also

ERROR(), MESSAGE()

Description

If SET ERRORVERSION is ON when an error occurs, instead of creating an *error.mem* file, an *error<version number>.mem* is created. The <version number> will start at 0001 for the first error file created and will be incremented with each subsequent error.

If the environment variable *DB_ERRORDIR* is not defined, then the error files are created in the current directory, otherwise the error files will be created in the specified directory. This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, ERRORVERSION is set ON. If a value of .F. is returned, ERRORVERSION is set OFF. By default ERRORVERSION is OFF.

Example

```
set errorversion on
```

Products

Recital Mirage Server, Recital Terminal Developer

SET ERRORWINDOW

Class

Error Handling and Debugging

Purpose

Enable or disable the use of error windows

Syntax

SET ERRORWINDOW ON | OFF | (<expL>)

See Also

SET TRACEWINDOW, DEBUG

Description

The SET ERRORWINDOW command is used to enable or disable the use of error windows. A window is an area of the screen designated for output and input. Windows are defined with the DEFINE WINDOW command, and are activated with the ACTIVATE WINDOW command. There is no limit to the number of windows you may define. An error window is a window that automatically activates itself and displays error messages, and a button marked *Confirm*. Pressing the [RETURN] key deactivates error windows. Error windows are specified by including the ERROR keyword with the DEFINE WINDOWS command.

When SET ERRORWINDOW is ON, error windows activate whenever an error occurs. When SET ERRORWINDOW is OFF, error messages display in the currently active window.

Example

```
set errorwindow on
```

Products

Recital Terminal Developer

SET ESCAPE

Class

Keyboard Events

Purpose

Toggle action of the interrupt key to halt execution of a program file

Syntax

SET ESCAPE ON | OFF | (<expL>)

See Also

ON ESCAPE, SET DOESCAPE, ON KEY, INKEY()

Description

If SET ESCAPE is ON, then pressing the interrupt key will cause the current program to be halted. If the current command being executed at the time of the interrupt key being pressed was entered in interactive mode, then the command is canceled.

If a program is executing, then the pop-up debugger is activated. If you choose the Cancel option, control will return to the '>' prompt. If you choose the Suspend option, execution of the program is suspended. You can then enter interactive commands at the '>' prompt, and resume operation of the program with the RESUME command. If you choose the Ignore option, then program execution continues as if no interrupt had occurred.

The SET DOESCAPE command can be used to cancel only the current command during the execution of a program. The interrupt key can be trapped with the ON ESCAPE command. By default, ESCAPE is ON.

Example

```
procedure escape
set message to "Operation canceled."
return to master
```

```
set escape on
do longjob
on escape do escape
do biggerjob
```

Products

Recital Terminal Developer

SET EXACT

Class

Environment

Purpose

Determines how comparisons between two character expressions are performed

Syntax

SET EXACT ON | OFF | (<expL>)

See Also

SET PCEXACT, TRIM(), LTRIM(), RTRIM(), SUBSTR(), LEFT(), RIGHT(), LPAD(), RPAD(), SOUNDEX()

Description

If SET EXACT is OFF, strings are compared up to the length of the shortest string. If SET EXACT is ON, the strings compared must be an exact match in both characters and length. By default, EXACT is OFF.

Example

```
set exact off
use patrons index events
// Lists all people beginning with "SMITH"
// SMITH, SMITHWAITE SMITHSON etc.
list all for name = "SMITH"
set exact on
// Lists only those people called "SMITH"
list all for name = "SMITH"
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET EXCLUSIVE

Class

Environment

Purpose

Determines whether or not table and index files are opened shareable

Syntax

SET EXCLUSIVE ON | OFF | (<expL>)

See Also

USE

Description

If SET EXCLUSIVE is ON when a table and its associated index files are opened, then the table and index files cannot be by another user until they are closed. If SET EXCLUSIVE is OFF when a table and its associated index files are opened, then the table and index files are shareable by multiple users. File and record locking to control concurrent updates to the table and index files is automatically enforced if EXCLUSIVE was OFF when the table was opened. The commands MODIFY STRUCTURE, PACK, REINDEX, INDEX ON (tags) and ZAP are the only commands that require a table to be opened exclusively. By default, EXCLUSIVE is ON.

Example

```
// Close today's payments
procedure close_day
set exclusive on
set view to accounts
if used()
    select payments
    set dcache to recount()
    set dcache on
    seek dtos(date())
    total on dtos(payment_date)+emp_no to (cdow(date));
    while payment_date = date()
    seek dtos(date())
    delete rest while payment_date = date()
    pack
else
    dialog box "Cannot open payments exclusively."
endif
set exclusive off
set view to payments
return
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FASTINDEX

Class

Indexing

Purpose

Quickly build temporary indexes

Syntax

SET FASTINDEX <ON | OFF>

See Also

INDEX ON

Description

The Recital DBMS supports two types of index creation: normal and fast indexing. When SET FASTINDEX is set ON, index files will be built faster at the expense of a well balanced b-tree, this means updates will take longer because the tree is re-balanced as it is being used. When SET FASTINDEX is OFF, indexes may take longer to build, but will be completely balanced when the INDEX command has finished. This is the preferred method for creating indexes. Regardless of the method used to create the index, as keys are added, modified and removed from the index, balancing of the b-tree will continue. The SET FASTINDEX command has no effect when using the REINDEX command.

CAUTION: The intended proper use of SET FASTINDEX is for the rapid creation of temporary indexes that are only meant to be used once and then discarded. It is not recommended for building a permanent index of any kind.

Example

```
set fastindex off
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FCACHE

Class

Performance and Optimization

Purpose

Enable or disable open file caching

Syntax

SET FCACHE ON | OFF | (<expL>)

See Also

CLEAR FCACHE, CLEAR PROGRAM, SET PCACHE, SET DCACHE, SET ICACHE

Description

If FCACHE is ON, any files that are opened are recorded in the 'open file cache '. When the file is closed, it is logically closed but not physically closed in the operating system. Further requests to open the file are satisfied from the 'open file cache', thereby speeding up program execution by removing an operating system action. The files are stored in the 'open file cache' until the process file limit is exceeded.

If FCACHE is OFF, closing files physically closes them in the operating system. If FCACHE is ON, the CLEAR FCACHE command must be issued as well as a CLOSE command in order to physically close files in the operating system. The default setting for FCACHE is OFF.

Example

set fcache on

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FIELDS

Class

Fields and Records

Purpose

Specify a list of fields that are accessible in a table

Syntax

SET FIELDS TO <field list> | ALL

SET FIELDS ON | OFF | (<expL>)

See Also

SET VIEW, FIELD(), FCOUNT()

Description

The SET FIELDS TO <field list> command defines a list of fields which are accessible from the currently selected table. The SET FIELDS TO ALL command allows all fields to be displayed. The <field list> is not used unless SET FIELDS is ON. The <field list> can only contain fields from the currently selected table. By default FIELDS is OFF, and all fields in the currently selected table are accessible.

Example

```
use patrons index events
set fields to date, event, name
set fields on
list for date<=date()
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FIELDVAL

Class

Screen Forms

Purpose

Specify the value to be entered in a field on a form

Syntax

SET FIELDVAL TO <expC>

See Also

@...GET, SET VALIDATE

Description

The SET FIELDVAL TO <expC> command allows modification of a field value that is being validated from data entered into a form. This command should be used in conjunction with the @...GET VALIDATE WITH command. If the VALIDATE WITH option is used on non-character variables, SET FIELDVAL TO can still be used, because the <expC> is automatically converted to the data type of the calling variable.

Example

```
procedure check_event
parameters data
if data = " "
    set fieldval to "BALLET"
endif
set validate on
return
```

```
@10,0 say "Event ";
get event validate with check_event
read
```

Products

Recital Mirage Server, Recital Terminal Developer

SET FILECASE

Class

Disk and File Utilities

Purpose

Allows case sensitive processing of file/directory names

Syntax

SET FILECASE <ON | OFF>

See Also

See Below

Description

When set on, file and directory processing is case sensitive. When set off, all filenames and directory names are set to lowercase. SET FILECASE ON must be active before issuing the COMPILE command to enable case sensitivity in compiled programs (dbo files). Default is OFF.

The following commands and functions are affected by SET FILECASE:

COMPILE	DO	SET DEFAULT	SET PATH
SET PRINTER TO	ADIR()	BINCREATE()	BINOPEN()
DIR()	FCREATE()	FDATE()	FILE()
FILECOUNT()	FOPEN()	FTIME()	IFILECOUNT()
MEMOREAD()	MEMOWRITE()	TEXTEDIT()	

Example

set filecase on

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FILETYPE

Class

Xbase Compatibility

Purpose

Table and index file creation in Xbase formats.

Syntax

SET FILETYPE TO [RECITAL | DBASE3 | DBASE4 | DB4 | FOXBASE | FOXPRO | FOXPLUS |
CLIPPER | VFP]

See Also

COPY, CREATE, EDITFIELD(), FILETYPE(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET
COMPATIBLE, SET EDITFIELD, SET FILECASE, SET INDEXEXT, SET MEMOEXT, SET PCEDIT,
SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET
PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS,
DB_SAMBA

Description

The SET FILETYPE TO command is used to specify a Replaceable Database Driver (RDD) to invoke when creating new database and index files. After issuing a SET FILETYPE TO command, all database, index and memo files created will be in the native Xbase format specified.

The Recital DBMS supports dynamic RDDs that allow transparent read/write access to database, memo and index files created in other Xbase languages. Access to these files is independent of the SET FILETYPE TO command.

Regardless of the file type used, the Recital Applications Data Dictionary (ADD) may still be used. When an Xbase database file is opened, the Recital RDD will look in the current directory for a file of the same name, but with a '.dbd' extension. If this file exists, it will be attached and all ADD attributes will be applied to the table.

The SET COMPATIBLE command has no effect on the format of files created in Recital products. By default SET FILETYPE TO uses native Recital file formats.

NOTES:

1. The dynamic RDDs are not available in Recital products for OpenVMS.
2. Access to other Xbase file formats via the RDDs requires extra system resources, due to the binary conversions required to access the data. This can result in performance degradation, particularly on large database files.

Example

```
use state.rdb
set filetype to clipper
copy to state_pc
use state_pc
index on STATE to state_pc
close
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer (None OpenVMS)

SET FILTER

Class

Table Organization

Purpose

Seclude records in a table that do not satisfy a certain condition

Syntax

```
SET FILTER TO [<condition>]  
SET FILTER ON | OFF | (<expL>)
```

See Also

SET RELATION, SET VIEW, FILTER()

Description

If SET FILTER TO <condition> is specified, records that do not satisfy the specified <condition> are automatically secluded. If SET FILTER TO is specified, without a <condition>, the current filter is removed allowing all records to be accessed. The filter <condition> is specific to each workarea, and is removed when a table is closed, or the workarea is re-used. However, the FILTER condition can contain references to fields in related tables. The FILTER() function can be used to save the current filter <condition>. The SET FILTER command enables or disables the filtering process. By default, FILTER is ON.

Example

```
use patrons  
set filter to event = "BALLET" and date < date()  
browse
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FIXED

Class

Numeric Data

Purpose

Determines whether a fixed number of decimal places are displayed for numeric output

Syntax

SET FIXED ON | OFF | (<expL>)

See Also

SET DECIMALS, ROUND()

Description

If SET FIXED is ON, then all numeric output will be displayed with the number of decimal places specified by the SET DECIMALS command. If no SET DECIMALS command has been issued, then the number of decimal places displayed will be 2. By default, FIXED is OFF.

Example

```
? 285.129 * 4.6
```

```
1311.5934
```

```
set decimals to 3
```

```
set fixed on
```

```
? 285.129 * 4.6
```

```
1311.593
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FKLABEL

Class

Screen Forms

Purpose

Toggle mapping of control keys to their function key equivalents in @...SAY commands

Syntax

SET FKLABEL ON | OFF | (<expL>)

See Also

@...SAY, DIALOG BOX, SET MESSAGE, MESSAGE

Description

The SET FKLABEL ON command enables mapping of control keys to their function key equivalents in @...SAY commands. Any control characters (e.g. ^C) embedded in text associated with @...SAY commands, will be translated into their equivalent function names. This command is therefore very useful for terminal independent displays. Mapping is always performed on messages written with SET MESSAGE and MESSAGE commands.

Example

```
set fklabel on
```

Products

Recital Terminal Developer

SET FORMAT

Class

Screen Forms

Purpose

Select a screen format for full screen forms based operations

Syntax

SET FORMAT TO [<.fmt filename> | (<expC>)]

See Also

SET AUTOFORMAT, SET DESCRIPTIONS, SET DEVICE, SET VIEW, @...GET, READ, SAVE SCREEN, RESTORE SCREEN, FMT()

Description

The SET FORMAT TO <.fmt filename> command selects a screen format which will be used with the full screen forms based commands: APPEND, CHANGE, EDIT, INSERT, QUERY and READ. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.fmt' is used. If the SET FORMAT TO command is issued without a <.fmt filename>, the current format file is deselected. The @ and SET commands are processed from the screen format file, other commands are ignored. The validity of the @ commands contained in the screen file is not checked until the form is activated with one of the full screen forms based commands.

The Screen Painter, initiated with the CREATE SCREEN command, automatically generates screen format files. Each workarea can have a separate format file specified. If no format file has been specified, and a full-screen forms-based command is executed, then a default form is generated.

The READ command does not handle user-defined forms in the same manner as the other forms based commands. The READ command executes the 4GL statement in the format (.fmt) file as if they were executed as part of a program module. The @...GETS can be exited by pressing [ENTER] on the last field only if SET READINSERT is ON and the last GET field is not a MEMO data type. Form level triggers are only executed once when the READ command is used to activate the form. Behavior of the READ command is not affected by the SET READEXIT command when used in conjunction with SET FORMAT TO and READ.

Note Format files can be nested. Whenever a menu bar is activated from a format file, the current state of the form is saved. The FMT() function can be used to save the name of the current format, another can then be opened, and afterwards the original can be re-opened.

Example

use demo
set format to demo

Products

Recital Mirage Server, Recital Terminal Developer

SET FORMSTATE

Class

Screen Forms

Purpose

Enable or disable Query Mode upon exiting a menu

Syntax

SET FORMSTATE ON | OFF | (<expL>)

See Also

SET UPDATE, SET QUERYMODE

Description

The SET FORMSTATE command is used to control the setting of Query | Update mode upon exit from a menu. When a menu is accessed from a work surface such as EDIT, the form is placed into Query Mode. If FORMSTATE is set ON, the mode of the form will be saved when the menu bar is accessed and then restored when the menu is exited. If FORMSTATE is OFF, then the form will always be placed into Query Mode upon exit from the menu. By default, FORMSTATE is turned ON in the main config.db file.

Example

```
// Force Query Mode upon menu exit  
set formstate off
```

Products

Recital Terminal Developer

SET FORMUPDATE

Class

Screen Forms

Purpose

Automatically write record changes to disk

Syntax

SET FORMUPDATE <ON | OFF>

See Also

CREATE SCREEN

Description

When set on, automatically writes changed records to disk when accessing menus on a form. Default is ON.

Example

set formupdate on

Products

Recital Terminal Developer

SET FULLPATH

Class

Environment

Purpose

Controls file specification display

Syntax

SET FULLPATH ON | OFF | (<expL>)

See Also

SET PATH, LIST STATUS, BASENAME(), DBF(), FMT(), NDX ()

Description

If FULLPATH is set ON, then the full file specification, including disk and directory, is displayed by commands and functions that display or return filenames. When FULLPATH is set OFF only the filename is displayed or returned. The SET FULLPATH OFF command can be used when programs whose functions return full file and path names are incompatible with programs whose functions only return the drive and the filename. The default for SET FULLPATH is OFF. The function BASENAME() can be used with FULLPATH to return only the filename when SET FULLPATH is ON.

Example

```
use state.rdb
```

```
? dbf()
```

```
state.rdb
```

```
set fullpath on
```

```
? dbf()
```

```
/usr/recital/UD/demo/state.rdb
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET FUNCTION

Class

Recital Terminal Developer Environment

Purpose

Program the terminal function keys

Syntax

SET FUNCTION <expN> TO <expC>

SET FUNCTION <expC1> TO <expC2>

See Also

SET CAPTURE, SET PCKEYS, REPLAY, MENU, @...GET, FKLABEL(), FKMAX()

Description

The SET FUNCTION TO command provides the capability of reprogramming the function keys of the terminal. The SET PCKEYS command must be ON prior to programming function keys. When specifying a function key to be programmed, SET FUNCTION accepts the function key number <expN>, or the function key label <expC>. The character string assigned to a function key can be up to 50 characters long and may contain only one command.

Reprogramming the function keys has no effect on their standard usage within full screen forms based operations. Terminal function keys can only be used at the interactive prompt.

Example

set function 2 to “modify command “
set function [PF1] to “run show users”

Products

Recital Terminal Developer

SET GATEWAY

Class

Gateways

Purpose

Establish a gateway to a Recital Database Gateway

Syntax

SET GATEWAY TO [<expC1>] [IN <workarea/alias>] [ALIAS <expC2>]

See Also

LOGIN, GATEWAY(), CONNECTED()

Description

The SET GATEWAY command is used to establish a connection to a Recital Database Gateway via the Recital Database Server. Each workarea can have a separate gateway established.

<expC1> is a character string that must be formatted in the following way:

servername@machinename:username/password-database.protocol

If <expC1> is not included with the SET GATEWAY command, the connection in that workarea will be detached.

An optional ALIAS <expC2> keyword can be used to specify an alias name for the workarea that is currently connected.

NOTE: For local ODBC data sources, a short format of <expC1> can be used, specifying only the data source name as configured in the ODBC Data Source Administrator.

NOTE: If the *protocol* is not specified, TCP/IP is assumed.

NOTE: The *database* entry when connecting to Oracle is not required. This entry can be used to pass hostname information when SQL*NET is being used. The information required differs depending on the version of SQL*NET:

SQL*NET 1

database = T:<node>:<SID>

SQL*NET 2

database = <service name>

Example

```
set gateway to "ora@sales:scott/tiger"  
// short format for local ODBC data source  
set gateway to "odbc:Northwind"
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET GCACHE

Class

Performance and Optimization

Purpose

Enable or disable shared table and index file caching

Syntax

SET GCACHE ON | OFF | (<expL>)

See Also

SET CACHELOAD, SET DCACHE, SET ICACHE, SET PCACHE

Description

The SET GCACHE (global cache) command allows table and index file caching on shared tables. The SET DCACHE and ICACHE commands cause table records and index files to be 'cached' in memory. Normally these commands only operate on tables that have been opened exclusively. When SET GCACHE is ON, the DCACHE and ICACHE commands handle caching on a distributed basis.

The number of records cached is specified with the SET DCACHE command. When SET GCACHE is ON, the number of records specified by the SET DCACHE command is reserved on a per user basis. Caching table records and index files accelerates only those I/O operations that access the cached records. Operations such as a sequential read of a table, or a random query, will not benefit from caching. The SET GCACHE command can be used in conjunction with the SET DCACHE and SET PCACHE commands to optimize performance.

The SET CACHELOAD command may be used with the DCACHE command to forcibly load the table and index cache into memory when the table is used. Although this can be a lengthy process, further access to the table is accelerated dramatically.

Example

```
// Open up payroll system
set cacheload on
set gcache on
set icache to 3000
set dcache to 1000
use payroll index pay_date, emp_code
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET HEADING

Class

Input/Output

Purpose

Determine the display of column titles above the output of certain commands

Syntax

SET HEADING ON | OFF | (<expL>)

SET HEADING TO SINGLE | DOUBLE | NONE

See Also

SET DESCRIPTIONS, LIST, DISPLAY, SUM, AVERAGE

Description

If SET HEADING is ON, then column titles will be displayed for the DISPLAY, LIST, SUM and AVERAGE commands. If SET HEADING is OFF, then no column titles will be displayed. If SET DESCRIPTIONS is ON, then the field description is displayed as the column title in place of the field name. By default, HEADING is ON.

The SET HEADING TO command controls the underlining of column headings in output files. The SINGLE or DOUBLE options will generate SINGLE or DOUBLE underlining on column headings. If NONE is specified then no underlining is generated for the column headings. When used in conjunction with the LIST | DISPLAY ... TO FILE, SET DESCRIPTION, SET PAGEWIDTH and SET PAGELength commands, SET HEADING provides the ability to produce ad-hoc reports. The default for SET HEADING is NONE.

Example

use patrons index events
list first 2 event, name off

event	name
-----	-----
BALLET	Webber
BALLET	Collins

set heading off
list first 2 event, name off
BALLET Webber
BALLET Collins

set pagelength to 60
set heading to single
use demo
list fields acc_prefix, acc_no, ord_value, name;
 to file accounts.txt;
 for ord_date = ("01/05/2000")

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET HELP

Class

Screen Forms

Purpose

Determines whether help messages are displayed

Syntax

SET HELP ON | OFF | (<expL>)

See Also

HELP, @...GET

Description

If SET HELP is ON, then Data Dictionary Help or @...GET...HELP messages will be displayed in the message line when a field gains input focus. By default, HELP is ON.

Example

```
@10,10 say "Event ";
      get event;
      help "Enter the event name."
set help on
read
```

Products

Recital Terminal Developer

SET HELPFILE

Class

Recital Terminal Developer Environment

Purpose

Specify an application specific help system

Syntax

SET HELPFILE TO <.hlm filename> | (<expC>)

See Also

HELP, SET HELP, SET HELPWINDOW, SET INSTRUCT, MENU HELPFILE, @...MENU HELPFILE, TEXTEDIT()

Description

The SET HELPFILE TO <.hlm filename> command allows you to bypass the main help system and specify your own help system for a particular application. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.hlm' is used. SET HELPFILE TO without an expression reselects the main help system.

The full path name of the help file must be specified.

Example

set helpfile to /usr/apps/help/accounts.hlm

Products

Recital Terminal Developer

SET HELPWINDOW

Class

Screen Environment

Purpose

Specify coordinate position of pop-up help window

Syntax

SET HELPWINDOW AT <expN1>,<expN2> TO <expN3>,<expN4> | DEFAULT

See Also

@...MENU HELPFILE, SET INSTRUCT, SET MEMOWINDOW

Description

The command SET HELPWINDOW specifies the coordinate position of the pop-up help window for @...MENU HELPFILE activation. The first coordinates define the upper left corner of the pop-up window, and the second set defines the lower right corner. If the columns specified for the help window are 0 and 79, then a 'navigation bar' is displayed at the bottom of the window. A 'navigation bar' is a highlighted display of available keys that aid the user to get help, page up, page down, or exit the help window. If SET HELPWINDOW AT DEFAULT is used, the screen coordinates are set to 1,0 TO 21,79.

Example

```
set helpwindow at 3,0 to 10,79
@2,0 menu "Examples";
    command "do examples";
    help "Do examples. Press ^C for help.";
    helpfile example.hlp
menu quit
```

Products

Recital Terminal Developer

SET HIDDENFIELD

Class

Fields and Records

Purpose

Specify the character to be displayed in hidden fields

Syntax

SET HIDDENFIELD TO [<expC>]

See Also

CREATE, MODIFY STRUCTURE, SET FIELDS

Description

The SET HIDDENFIELD TO <expC> command allows you to specify the character that will be displayed in hidden fields. Hidden fields are fields that are not visible to users for the purpose of disallowing read or write privileges. Fields are defined as hidden in the Application Data Dictionary (ADD). The ADD may be accessed from the CREATE or MODIFY STRUCTURE work surfaces where the <PROTECTION> menu contains the <HIDDEN> option for defining fields as hidden. Access to this option can be disabled from the ADMIN option from the <SECURITY> menu in the CREATE or MODIFY STRUCTURE work surfaces. Using the SET HIDDENFIELD TO command without specifying a character <expC> returns the character to the default ('?').

Example

set hiddenfield to '#'

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET HIGHLIGHT

Class

Recital Terminal Developer Environment

Purpose

Dim background intensity of the screen

Syntax

SET HIGHLIGHT ON | OFF | (<expL>)

See Also

SET BORDER, SET BLINK, SET INTENSITY, SET UNDERLINE, SET DELIMITERS, @...GET, READ, EDIT, BROWSE, APPEND, BLINK(), BOLD(), REVERSE(), UNDERLINE()

Description

When HIGHLIGHT is set ON, background intensity of the screen is dimmed when a pop-up is activated. All bold characters on the screen are dimmed while the menu is active, and then restored to their original intensity when the screen is restored. This has been added to distinguish the pop-up object from the screen on which it overlays. If HIGHLIGHT is OFF, bold characters remain bold when a menu or dialog box appears on the screen. By default, HIGHLIGHT is OFF.

Example

```
set highlight on
set blink on
use demo
edit
// Press the [HELP] key
```

Products

Recital Terminal Developer

SET HISTORY

Class

Error Handling and Debugging

Purpose

Determines whether commands are stored in a command history buffer

Syntax

SET HISTORY TO [<expN>] | FILE <.his filename> | (<expC>)
SET HISTORY ON | OFF | (<expL>)

See Also

DISPLAY HISTORY, DO, LIST HISTORY, SET COMPILE, SET DEVELOPMENT, SET DOHISTORY, SET HISTPATH, DB_SAVEHISTORY

Description

The command history buffer holds the most recent commands that have been executed from the command prompt. To inspect the command history, use the LIST HISTORY or DISPLAY HISTORY commands. If KBEDIT is set ON, the [CURSOR UP] and [CURSOR DOWN] keys can be used to scroll through and select a previous command for re-execution. The command can be edited before being re-executed. By default HISTORY is ON and 20 command lines are stored.

If SET DOHISTORY is also ON, then commands executed from interpreted program files will also be stored in the command history buffer. Program files are run interpreted, rather than compiled, when SET COMPILE is OFF and SET DEVELOPMENT is OFF.

Command history buffers are saved between sessions in the 'command.his' file. Subsequent sessions in the same directory will be able to access earlier history buffers. This functionality is only available if the environment variable / symbol DB_SAVEHISTORY is set to true.

TO <expN>

The SET HISTORY TO <expN> command specifies the maximum number of command lines that will be stored.

TO FILE

The TO FILE clause enables you to record your history in the specified <.his filename>. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. If SET DOHISTORY is also ON then commands executed from interpreted program files will also be stored in the command history. Each line that is executed is placed into the <.his filename>, next to the name, stack number and line number of the executing program. If SET HISTPATH is ON, then the commands also include the current directory name, e.g.

```
/home/recital/test/:keyboard(0:1): do myprog  
/home/recital/test/:myprog(1:1): do subprog  
/home/recital/test/:subprog(2:1): open database southwind
```

Example

```
// Example using config.db file
set history on
set histpath on
set history to file myhist
set dohistory on
// End of config.db
```

```
// Command prompt example
```

```
set history to 100
set history on
dir
use patrons index events
list history
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET HISTPATH

Class

Error Handling and Debugging

Purpose

Determines whether the current directory name is added to commands stored in the command history file

Syntax

SET HISTPATH ON | OFF | (<expL>)

See Also

DISPLAY HISTORY, DO, LIST HISTORY, SET COMPILE, SET DEVELOPMENT, SET DOHISTORY, SET HISTORY

Description

The SET HISTPATH command determines whether the current directory name is added to commands stored in the command history file. A command history file is created when SET HISTORY is ON and set to output the history buffer to a file. This is achieved using the SET HISTORY TO FILE command. If SET DOHISTORY is also ON then commands executed from interpreted program files will also be stored in the command history file. Each line that is executed is placed into the file, next to the name, stack number and line number of the executing program, e.g.

```
/home/recital/test/:keyboard(0:1): do myprog  
/home/recital/test/:myprog(1:1): do subprog  
/home/recital/test/:subprog(2:1): open database southwind
```

Example

```
// Example using config.db file  
set history on  
set histpath on  
set history to file myhist  
set dohistory on  
// End of config.db
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET HOURS

Class

Recital Terminal Developer Environment

Purpose

Change time display to 12 or 24 hours

Syntax

SET HOURS TO [12 | 24]

See Also

AMPM(), CDOW(), CMONTH(), CTOD(), CTOT(), DATE(), DATETIME(), DAY(), DAYS(), DMY(), DOW(), DTOC(), DTOM(), DTOS(), DTOV(), ELAPTIME(), EMPTY(), EPOCH(), GOMONTH(), HOUR(), HOURS(), LTOS(), MDY(), MINUTE(), MINUTES(), MONTH(), MTOD(), MTOS(), QUARTER(), SEC(), SECONDS(), SECS(), STOD(), STR(), TIME(), TIMESTAMP(), TSTRING(), TTOC(), TTOD(), TYPE(), VAL(), VALIDTIME(), VTOD(), YEAR(), SET CENTURY, SET DATE, SET EPOCH, SET MARK, SET SECONDS, SET VAXTIME

Description

The SET HOURS TO command changes the system clock to a 12 hour or a 24 hour display. If the optional qualifier 12 | 24 is not specified, then the clock is set to the default. The default setting is 12 hours. This command also affects the display of datetime values, determining whether hours are shown in 24 hour format or in 12 hour format with AM | PM postfix.

Example

```
set hours to 24
```

Products

Recital Terminal Developer

SET IBLOCK

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET IBLOCK TO <expN>

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

set iblock to 4

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ICACHE

Class

Performance and Optimization

Purpose

Specify number of buffers to use for index files

Syntax

SET ICACHE TO [<expN>] [OPTIMUM]

See Also

SET CACHELOAD, SET DCACHE, SET ICACHE, SET GCACHE, SET PCACHE

Description

The SET ICACHE TO <expN> command specifies the number of index file buffers to be used. The icache buffer is specific to each index file that is opened. The larger the number of ICACHE buffers, the better the performance of FIND and SEEK operations on large tables. The ICACHE can also accelerate the INDEX, REINDEX, TOTAL and SORT commands. If the OPTIMUM keyword is specified, then the optimum size for index caching will automatically be calculated. For manual settings, the <expN> can be calculated with the ICACHE() function.

There is no benefit in having a large ICACHE with index files that are shareable. The SET ICACHE command has no effect if a table is shareable, unless SET GCACHE is set ON. When SET GCACHE is ON, SET DCACHE and SET ICACHE operate on a distributed basis. The SET ICACHE command can be used in conjunction with the SET PCACHE, and the SET DCACHE commands to optimize performance. The value <expN> specified with the SET ICACHE command is the number of disk blocks to be buffered in memory. When a request is made to read a particular disk block from an index file, the ICACHE is first inspected to see if the block is already in memory. If it is, then the block is merely copied from memory. If the block is not in memory, the required block is read from the index file, and placed it in the ICACHE. The LIST STATUS command displays the current setting of ICACHE of each individual index. By default, ICACHE is set to 20, the minimum value is 2.

Example

```
set icache to 100
use patrons index events, dates, names
// Another example
use accounts index acc_prefix, paid_date
set icache to reccount()
reindex
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET INDEX

Class

Indexing

Purpose

Open a list of index files associated with the active table

Syntax

SET INDEX TO [<index list>] | [ORDER <.ndx file> |<.dbx tag name>[OF <.dbx file>]] [ADDITIVE]

See Also

CLOSE INDEX, INDEX, REINDEX, USE, MDX(), TAG(), TAGCOUNT(), TAGNO(), SET COMPATIBLE, SET INDEXEXT, SET ORDER, SET VIEW

Description

The SET INDEX TO command is used to open single index files and /or multiple .dbx index files for a table. A maximum of twenty .ndx and .dbx files can be associated with a table at any one time. All of the open index files associated with a table are automatically updated when any modifications are made to the table.

If there is no production .dbx file, the first index in <index list> will be the master index, that is, the records are in order by this index expression. If there is a production index associated with the .dbf, the first tag in the production .dbx will be the master index. The ORDER qualifier may be used to specify a different order to be active after the indexes have been opened. If the tag name specified with the ORDER qualifier is used in more than one .dbx file, the OF qualifier can be used to specify the appropriate .dbx file name. The SET ORDER TO command can be used to select a master index from the currently active index files.

If the optional ADDITIVE keyword is specified, the indexes included in <index list> are opened in addition to any currently open indexes. If ADDITIVE is not specified or if the SET INDEX TO command is issued without an <index list>, any currently open indexes in the active workarea are closed.

Example

```
use patrons
set index to names, events
seek "smith"
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET INDEXEXT

Class

Xbase Compatibility

Purpose

Specify file extension for index filenames

Syntax

SET INDEXEXT TO [<expC1>[, <expC2>]]

See Also

INDEX, REINDEX, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEX, SET MEMOEXT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET INDEXEXT TO command specifies the extension that will be used for index files.

Parameter	Description
<expC1>	Specifies the extension for single index files. If omitted, the single index file extension is set to the default extension 'ndx'.
<expC2>	Specifies the extension for multiple (tag) index files.

The default index file extensions are also affected by the SET COMPATIBLE and SET CLIPPER commands.

Example

```
set indexext to ".ntx", ".cdx"  
index on name to names  
use name index names
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET INKEYDELAY

Class

Keyboard Events

Purpose

Enable or disable delays between checks for keys pressed in the INKEY() function

Syntax

SET INKEYDELAY ON | OFF | (<expL>)

See Also

SET PCKEYS, SET MESSAGE TO, MESSAGE, @...MENU, MENU, WAIT, INKEY()

Description

SET INKEYDELAY ON controls the wait period for the INKEY() function. The INKEY() function returns a zero if no key has been pressed, or the ASCII value of the key which has been pressed. When INKEYDELAY is ON, a check is made once a second for a keystroke. If INKEYDELAY is OFF, the check is made continually. The default setting for INKEYDELAY is OFF. If INKEYDELAY is set OFF, then DO WHILE loops that wait for user input with the INKEY() function may be very CPU intensive.

Example

```
set inkeydelay on
do while inkey()=0
    //...
enddo
```

Products

Recital Mirage Server, Recital Terminal Developer

SET INSTRUCT

Class

Recital Terminal Developer Environment

Purpose

Enables or disables pop-up help

Syntax

SET INSTRUCT ON | OFF | (<expL>)

Description

If INSTRUCT is set ON, then the pop-up help windows which have been set up with the @...MENU HELPFILE or MENU HELPFILE command can be activated. When INSTRUCT is set OFF, these help facilities are disabled. By default, SET INSTRUCT is OFF.

See Also

@...MENU HELPFILE, SET HELPWINDOW TO, MENU HELPFILE

Example

```
procedure main
// Procedure to display main menu
//...
return

set instruct on
menu helpfile main.hlp format main
```

Products

Recital Terminal Developer

SET INTENSITY

Class

Screen Forms

Purpose

Determines whether fields displayed in forms will be highlighted in reverse video

Syntax

SET INTENSITY ON | OFF | (<expL>)

See Also

SET UNDERLINE, SET DELIMITERS, @...GET, READ

Description

If SET INTENSITY is ON, fields will highlighted in reverse video during full screen forms based operations, such as APPEND, CHANGE, EDIT, INSERT, QUERY and READ. By default, INTENSITY is ON.

Example

```
set intensity off
set underline on
use patrons index events, dates, names
change
```

Products

Recital Terminal Developer

SET JOURNAL

Class

Table Organization

Purpose

Enable after image journaling and audit trail for the active table

Syntax

SET JOURNAL TO [<.dbj filename> | (<expC>)]

SET JOURNAL ON | OFF | (<expL>)

See Also

COPY FILE, RECOVER, SET MEMOEXT, SET MEMOJOURNAL

Description

The SET JOURNAL TO <.dbj filename> associates a transaction journal file with the active table. If the specified journal file does not exist, it will be created. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.dbj' is used. The <.dbj filename> is a standard table. It contains seven fields that are specific to a journal file, followed by the first 249 fields of the associated table.

The first seven fields in the journal are:

Field	Type	Display	Storage	Description
AUD_DATE	Date	8 10	4	The date on which the transaction was performed.
AUD_TIME	Character	8	8	The time at which the transaction was performed, in the format HH:MM:SS.
AUD_TERM	Character	12	12	The name of the terminal from which the transaction was performed
AUD_UID	Short	5	2	The ID of the user who performed the transaction.
AUD_GID	Short	5	2	The group ID of the user who performed the transaction.
AUD_CMD	Short	4	2	The command number of the transaction performed from the command table below
AUD_RECNO	Integer	7	4	The record number in the associated table which the transaction was performed on.

Command	Number
DELETE	14
RECALL	36
REPLACE	41
BROWSE	6
CHANGE	8
EDIT	17
INSERT	26
APPEND	5
READ	35

The SET JOURNAL TO command without a <.dbj filename> specified closes the active journal file. The SET JOURNAL ON | OFF command enables or disables transaction journaling. This command is primarily used in applications where journaling can be disabled for a certain class of operations. The journaling features are mainly used with shared tables. It should be noted that there is an overhead in enabling transaction journaling, as records updated in a table are also written to the journal file.

When records are appended into a journal file, locking is automatically performed so that multiple users can update the journal concurrently. The associated table must be opened shareable for this to occur. Each table can have a journal file associated with it. Since journal files are standard Recital tables, you can use standard Recital commands such as the REPORT command to print audit trails, transaction logs, etc.

The RECOVER command works in conjunction with the SET JOURNAL TO <.dbj filename> command to recover after a fatal disk error. See the RECOVER command for full details. For RECOVER to work successfully, all journal files should be removed with the DELETE or ZAP commands after making backups of the tables.

When specifying a journal file, it is recommended that the journal file is stored on a different disk than that which the table is stored on, so that if a fatal disk error occurs, then the journal file will not be lost along with the table. By default, JOURNAL is OFF.

NOTE: Only the first 249 fields of a table can be journalled: subsequent fields are ignored.

Example

```
// Open journal file
set exclusive off
use payroll
set journal to payroll
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET KBEDIT

Class

Keyboard Events

Purpose

Provide command line recall and editing facilities

Syntax

SET KBEDIT ON | OFF | (<expL>)

See Also

SET HISTORY

Description

The SET KBEDIT ON command provides command line recall and editing facilities. If SET HISTORY is ON, commands can be recalled from history using the [CURSOR UP] and [CURSOR DOWN] keys. Previous command lines held in history can be edited in the same way as fields on forms. The SET HISTORY TO <expN> command specifies the number of commands that will be held in history.

Example

```
set kedit on
set history to 10
```

Products

Recital Terminal Developer

SET KEY TO

Class

Keyboard Events

Purpose

Map a menu option to a control or function key in a form

Syntax

SET KEY TO <expN>

See Also

SET KEY...TO, CTRL(), @...MENU, @...GET, KEYBOARD

Description

The SET KEY TO <expN> command is used in custom screen formats to map a menu option to a function or control key.

A screen format file, which is activated with the SET FORMAT TO <.fmt filename> command, can contain a form definition using @...SAY...GET commands, and can also contain a menu definition using @...MENU commands. When the form is activated by one of the full screen forms based commands APPEND, CHANGE, EDIT, INSERT, QUERY or READ, the form is displayed on the screen and data input mode is entered. If the [MENUBAR] key is pressed, then any @...MENU commands which have been defined in the screen format file are activated. You can then specify the SET KEY TO <expN> command as the command on the @...MENU command.

If you issue the SET KEY TO -1 command, activated from a menu selection, the menu will be exited. The CTRL() function is often used with the SET KEY TO command. The Screen Painter, initiated with the CREATE SCREEN command, allows menu items to be built into forms.

NOTE: It is important to distinguish this command from the following command (SET KEY <expN> TO <procedure name>), which has a different purpose. An alternative to the SET KEY command is to use the KEYBOARD command.

Example

```
// Dataentry.fmt
@0,0 say "Next"
@0,0 menu "Next";
    command "set key to ctrl('n')"
@0,5 say "Prev"
@0,5 menu "Prev";
    command "set key to ctrl('r')"
// The form would now be defined
return
```

```
use patrons index events, dates, names
set format to dataentry
change
```

Products

Recital Mirage Server, Recital Terminal Developer

SET KEY...TO

Class

Keyboard Events

Purpose

Define hot keys

Syntax

SET KEY <expN> TO <procedure name>

See Also

SET KEY, INKEY(), READKEY(), SET PCKEYS, PROCNAME(), PROCLINE(), READVAR(), PRINTSCREEN()

Description

The SET KEY...TO command enables a user, running an application, to press a key which will cause execution of the specified procedure whenever the program is waiting for keyboard input. All keys, excluding a through z and 0 through 9, may be used.

NOTE: It is important to distinguish this command from the previous command (SET KEY TO <expN>) which has a different purpose.

On completion of the procedure, execution of the application continues from the point at which it left. Any valid Recital command can be used in the procedure. The <expN> is the INKEY() value of the key and <procedure name> is the procedure to be executed when the key is pressed. The executed procedure specified in the SET KEY...TO command is passed three parameters with the values of PROCNAME(), PROCLINE(), and READVAR() respectively.

Example

```
procedure showlist
save screen
aliasnam = select()
use shows in workarea()
display all
select &aliasnam
restore screen
return
```

```
set key 28 to showlist
accept "Enter Show Name (F1 for Help)" to event
```

Products

Recital Mirage Server, Recital Terminal Developer

SET KEYWORD

Class

Screen Environment

Purpose

Specify keyword settings for BROWSE or EDIT for the current workarea

Syntax

SET KEYWORD OF BROWSE | EDIT TO [<keywords>]

See Also

SET COMMAND, BROWSE, EDIT, CHANGE

Description

The SET KEYWORD command can be used to specify the keyword settings for EDIT or BROWSE on a per workarea basis. The keywords specified are active in the workarea that was current when the command was issued. Each subsequent time that a BROWSE or EDIT is To reset BROWSE or EDIT to the default behavior, issue the command without specifying <keywords>. For details of the keywords for BROWSE or EDIT, please see the individual commands.

Example

```
select 1
set keyword of browse to nomenu nodelete
set keyword of edit to noedit
select 2
set keyword of browse to lock 1
```

Products

Recital Terminal Developer

SET LANGUAGE

Class

Screen Environment

Purpose

Specifies language

Syntax

SET LANGUAGE TO <language-type>

See Also

COLLATE()

Description

The SET LANGUAGE TO command specifies the language to be used in the COLLATE() function and also the collating sequence for SORT operations. The COLLATE() function recognizes foreign characters and places the specified character expression correctly within the existing sequence. On execution of the SET LANGUAGE TO command, the appropriate language translation table is loaded from the 'terminals' directory. The following languages are supported:

- AMERICAN
- BELGIAN
- BRITISH
- CHINESE
- DANISH
- DUTCH
- FINNISH
- FLEMISH
- FRENCH
- GERMAN
- ITALIAN
- JAPANESE
- NORWEGIAN
- RUSSIAN
- SPANISH
- SWEDISH

The character translation table for the specified language should be specified in the 'terminals' directory in a filename equivalent to:

<terminal-type>-<first-3-characters-of-language>.ncs

For example, the german translation table for the VT200 terminal can be found in the file: vt200-ger.ncs.

Example

```
set language to french
```

```
use customers
```

```
index on collate(custname) to frenchcust
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET LDCHECK

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET LDCHECK ON | OFF

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

```
set ldcheck on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET LIBRARY

Class

Environment

Purpose

Load an API library file built with the Recital SDK

Syntax

SET LIBRARY TO [<library filename> | (<expC>) [ADDITIVE]]

See Also

CLOSE PROCEDURE, DO, FUNCTION, LINK, LIST PROCEDURE, PARAMETERS, PROCEDURE, RELEASE LIBRARY

Description

The SET LIBRARY TO < library filename> command opens the specified API procedure library file, scans the contents of it, and records the names and positions of the procedures defined within it. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. You can place as many procedures as you want in an API procedure library file.

If the optional ADDITIVE keyword is specified then any API procedures that are already open are left open and the new API procedure library is added. The SET LIBRARY TO command, without any filename specified, closes all active API procedure library files. A closed library file discards any knowledge of where the procedures within reside. The RELEASE LIBRARY <library filename> command can be used to close an individual API library file.

The SET LIBRARY and RELEASE LIBRARY commands only affect API procedure library files, not Recital/4GL procedure library files: these are handled by the SET PROCEDURE and CLOSE PROCEDURE commands.

The active API procedures and functions can be listed with the LIST or DISPLAY PROCEDURE commands.

For full details on using the Recital SDK, please see the SDK documentation.

Example

```
// Open Samples.so API procedure library
set library to /usr/recital/UnixDeveloper/sdk/lib/Samples.so
// Open pdf.so procedure library without closing active libraries
set library to /usr/recital/UnixDeveloper/sdk/lib/pdf.so additive
// Close pdf.so API procedure library
release library /usr/recital/UnixDeveloper/sdk/lib/pdf.so
// Close all active API procedure library files
set library to
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET LOCKWAIT

Class

Environment

Purpose

Enable or disable automatic record locking

Syntax

SET LOCK ON | OFF | (<expL>)

SET LOCKWAIT ON | OFF | (<expL>)

See Also

FLOCK(), LOCK(), LOCKF, LOCKR, RLOCK(), UNLOCK

Description

The SET LOCK[WAIT] command determines locking behavior when you attempt to gain UPDATE access to a record which is already locked by another user. This command is only effective with shared tables.

With SET LOCK[WAIT] ON, commands such as REPLACE will wait until a lock can be obtained. With SET LOCKWAIT OFF, an error will be generated, telling you that the operation cannot take place as the 'Record is in use by another user'. In the forms system, both settings give you the opportunity to either wait for the record to be unlocked by the other user and your lock to be granted, or, to access the record in QUERY mode. To switch from 'Query' mode to 'Update' mode in a form, press the [UPDATE MODE] key. By default, LOCK is ON.

Example

use patrons index events, dates, names
set filter to event = "HAMLET"
browse

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET LOCKTYPE

Class

Manual Locking

Purpose

Determines whether locking strategies should be optimistic or pessimistic

Syntax

SET LOCKTYPE TO OPTIMISTIC | PESSIMISTIC

See Also

CHANGE(), GATHER, READ, REPLACE

Description

The SET LOCKTYPE set command is used to determine the locking strategy when a record is updated via temporary field placeholders such as memory variables or array elements. The default Recital behavior is SET LOCKTYPE TO PESSIMISTIC. With LOCKTYPE set to PESSIMISTIC, the record must be locked (RLOCK() or LOCKR) throughout the transaction to ensure that it cannot be modified by another user before the update is made. Without a lock being placed, the state of the data cannot be guaranteed.

With LOCKTYPE set to OPTIMISTIC, then prior to a REPLACE or GATHER operation the record is compared to the buffer from the previous read. If the two differ, an error is generated and the update does not take place. The CHANGE() function can be used to check whether the record has been modified before issuing the REPLACE or GATHER commands.

Example

```
set locktype to optimistic
use customer
store automem
@1,1 get m.name
@2,1 get m.address
@3,1 get m.state
read
if not change()
    replace customer.name with m.name,;
    customer.address with m.address,;
    customer.state with m.state
endif
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MACKEY

Class

Keyboard Events

Purpose

Enable or disable keyboard macros.

Syntax

SET MACKEY ON | OFF | (<expL>)

See Also

SET CAPTURE, REPLAY

Description

The SET MACKEY command enables or disables interactive keyboard macro recording and playing from within a form. Keyboard macros are keystrokes that have been recorded and stored in an ASCII file. By default, macro files have a '.kbm' extension.

The setting of MACKEY does not have any affect on the use of the SET CAPTURE and REPLAY commands.

When SET MACKEY is OFF, keyboard input recording/replaying is disabled. When SET MACKEY is ON, you may record and replay keyboard macros.

To begin capturing keystrokes, hold down the [CONTROL] key and press {O}. Release both keys and press {c}. A dialog box will be displayed asking for a filename to save keystrokes to. At this point capture mode is active. While in capture mode, 'Cap' is displayed in the status bar. After pressing the keystrokes that you wish to capture, hold down the [CONTROL] and {O} keys, release them, and then press {c}.

Macro files are replayed by selecting them from the Macros Menu. To activate the Macros Menu, hold down the [CONTROL] and {O} keys, release them and press {r}.

Example

```
set view to accounting
set mackey on
edit
```

Products

Recital Terminal Developer

SET MACROS

Class

Environment

Purpose

Enable or disable & macro substitution

Syntax

SET MACROS ON | OFF | (<expL>)

See Also

&, TEXT, SET EMACROS

Description

If SET MACROS is ON, then Recital substitutes & macros when command lines are read. If SET MACROS is OFF, then no macro substitution is performed. This command can be used to marginally accelerate program execution. Also, macros are normally substituted in the TEXT ... ENDTEXT command. Setting MACROS OFF disables this. See the & function for more details concerning macro substitution. By default MACROS is ON, so macro substitution is performed.

Example

```
set macros off
text
```

The & command is used to perform
macro substitution of the contents
of memory variables.

```
endtext
set macros on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MAIL

Class

Recital Terminal Developer Environment

Purpose

Trap and display OpenVMS mail messages.

Syntax

SET MAIL ON | OFF | (<expL>)

See Also

ON MAIL, MAIL()

Description

The SET MAIL command controls the handling of OpenVMS mail messages. When SET MAIL is ON, OpenVMS mail messages will be trapped and displayed in a pop-up window. The window clears when the user presses the [RETURN] key. When SET MAIL is OFF, mail messages are ignored.

This command has no affect on Operating Systems other than OpenVMS.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned MAIL is set ON. If a value of .F. is returned MAIL is set OFF.

Example

```
set mail on
```

Products

OpenVMS Developer

SET MAPCHAR

Class

Input/Output

Purpose

Define an alternate character to map CHR(0)

Syntax

SET MAPCHAR TO <expN>

See Also

CHR(), ???, ASC()

Description

The SET MAPCHAR TO command allows the user to define an alternate character to map CHR(0) for output processing. Whenever the CHR(0) function is used, or the ??? “{NULL}” command is used and MAPCHAR is set to a value other than 0, the CHR(0) character is translated into the character specified by the decimal value in <expN>. On output any occurrences of the character specified by <expN> will be translated back to CHR(0). Setting MAPCHAR TO 0 disables character mapping. The default setting is 0.

Example

```
? chr(64)
@
set mapchar to 64
? chr(0)
@
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MARGIN

Class

Printing

Purpose

Adjust the left-hand margin for all printed output

Syntax

SET MARGIN TO <expN>

See Also

CREATE REPORT, SET PRINT, SET PRINTER

Description

The SET MARGIN TO <expN> command adjusts the left margin for all printed output. The number <expN> is the number of characters to leave blank in the left margin. By default, the margin is 0.

Example

```
set margin to 10
use patrons index events
set printer to \\spooler
report form patrons to print
set printer to
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MARK

Class

Date and Time Data

Purpose

Change the separator character in dates

Syntax

SET MARK TO [<expC>]

See Also

DATE(), CTOD(), DTOC(), DTOS(), STOD(), DTOV(), VTOD(), DMY(), MDY(), SET DATE

Description

The SET MARK TO command changes the separator character in dates to the character specified in expression <expC>. If character expression <expC> evaluates to more than one character, the SET MARK TO command only uses the first character in the string. The default separator in dates is determined by the setting of the SET DATE TO command. If the optional <expC> is not used, the date separator is set back to the default.

Example

set mark to “.”

?date()

05.05.2000

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MAXDBO

Class

Environment

Purpose

Specify maximum compiled program size

Syntax

SET MAXDBO TO (<expN>)

See Also

SET COMPILE, SET DEVELOPMENT, COMPILE, DO

Description

The SET MAXDBO set command is used to specify the maximum compiled program size. The expression <expN> should be set to 4x the required maximum compiled file size in kilobytes. The default value is 256, giving a maximum compiled file size of 65536 bytes. If the MAXDBO setting is not sufficiently high, then an error will occur when a program is compiled manually or automatically. Automatic compilation takes place if SET COMPILE and SET DEVELOPMENT are ON when a program is run.

Example

```
set maxdbo to 512
set compile on
do main_prog
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MBLOCK

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET MBLOCK TO <expN>

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

set mblock to 1

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MCONFIRM

Class

Screen Environment

Purpose

Set up requirement for the [RETURN] key to be pressed following a menu selection

Syntax

SET MCONFIRM ON | OFF | (<expL>)

See Also

SET CONFIRM, @...MENU, MENU, SET BELL, DO

Description

The command SET MCONFIRM ON requires the [RETURN] key to be pressed after a selection has been made from any menu or dialog box. SET MCONFIRM works consistently throughout all menu bars, pulldown menus and dialog boxes. The menu bars of the Screen Painter, BROWSE, CREATE and MODIFY STRUCTURE work surfaces operate as pull down menus when MCONFIRM is OFF. The selected option is executed if the first key of the option is entered or if the option is highlighted. When MCONFIRM is ON, selected options do not execute until the [RETURN] key is pressed. Menus constructed with the @...MENU...PULLDOWN command are affected by the SET MCONFIRM command. By default, MCONFIRM is OFF.

Example

```
set mconfirm on
do menu_opt
```

Products

Recital Terminal Developer

SET MEMOCLEAR

Class

Recital Terminal Developer Environment

Purpose

Allows memo pads to refrain from automatic saving and restoring of the screen

Syntax

SET MEMOCLEAR ON | OFF | (<expL>)

See Also

MEMOSAY(), SET PRERECORD TO, SET MEMOCLEAR, SET MEMOWINDOW, SET MEMOWIDTH, MEMOWRITE(), MEMLINES(), MLINE(), MEMOEDIT(), TEXTEDIT(), MLCOUNT(), MEMOLINE(), MEMOTRAN(), HARDCR(), MEMOREAD()

Description

When MEMOCLEAR is set ON, pop-up memo note pads are automatically saved and restored. If MEMOCLEAR is set OFF, then pop-up memo note pads refrain from automatic saving and restoring of the screen. The default setting for MEMOCLEAR is ON.

Example

```
procedure saymemo
  memosay(details,2,41,10,79, "Customer Details")
return
```

```
use accounts
set memoclear off
set form to details
set prerecord to saymemo
edit
```

Products

Recital Terminal Developer

SET MEMOEXT

Class

Memos

Purpose

Specify the file extension for memo files

Syntax

SET MEMOEXT TO <expC>

See Also

EDITFIELD(), FILETYPE(), INDEXEXT(), MEMLINES(), MEMOEDIT(), MEMOLINE(), MEMOREAD(), MEMOWRITE(), MLCOUNT(), MLINE(), TEXTEDIT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOWIDTH, SET MEMOWINDOW, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, SET WP, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET MEMOEXT command is used to specify file extensions for memo files. If <expC> is omitted then the memo extension is set to the default extension '.dbt.' This command is useful for opening memo files that have been created with the CALENDAR MENU SCHEDULE command.

Example

```
calendar menu schedule file yyy
set indexext to ".sdx"
set memoext to ".sdt"
use yyy.sdb index yyy
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MEMOFORMAT

Class

Memos

Purpose

Enable or disable memo formatting during and after editing operations.

Syntax

SET MEMOFORMAT ON | OFF | (<expL>)

See Also

SET WP TO, SET MEMOWINDOW, SET MEMOWIDTH, MEMOEDIT(), MEMOSAY(), MEMLINES(), MEMOWRITE(), MLCOUNT(), MLINE(), TEXTEDIT(), MEMOLINE(), MEMOREAD(), MEMOTRAN(), HARDCR(), SET PRERECORD, SET MEMOCLEAR, SET MEMOWINDOW, SET MEMOWIDTH

Description

If MEMOFORMAT is ON, the memo field is automatically formatted when it is saved. All the lines are word-wrapped into the field, and carriage returns are removed. Hard carriage returns may be retained by toggling Insert Mode ON before pressing the [RETURN] or [ENTER] key. When MEMOFORMAT is set OFF, this allows *preformatted* text to be held in the memo without it being reformatted as it is used. All spaces and carriage returns are stored in the memo field.

WARNING: Once a memo is saved with MEMOFORMAT set ON, all the formatting characters are removed. This command should always be left ON or OFF, not toggled. The default for SET MEMOFORMAT is OFF

Example

```
procedure edit_memo
set memoformat off
memoedit(comp_hist,5,10,20,70,(getuid())>100),;
    "Company History")
return
```

Products

Recital Terminal Developer

SET MEMOJOURNAL

Class

Memos

Purpose

Cause memo fields to be included in after image journaling

Syntax

SET MEMOJOURNAL ON | OFF | (<expL>)

See Also

SET JOURNAL, SET MEMOEXT, RECOVER

Description

The SET MEMOJOURNAL command causes memo fields to be journalled when journaling is set on a table.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, MEMOJOURNAL is set ON. If a value of .F. is returned, MEMOJOURNAL is set OFF. By default SET MEMOJOURNAL is OFF.

Example

```
use mytable
set memojournal on
set journal on
set journal to /journals/mytabj
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MEMOSOFTCR

Class

Memos

Purpose

Add line feeds to memo text lines when reading from and writing to text files.

Syntax

SET MEMOSOFTCR ON | OFF | (<expL>)

See Also

MEMOREAD(), MEMOWRITE()

Description

The SET MEMOSOFTCR command causes Line Feed characters to be output after each line of text in the memo, when reading/writing a memo field from/to an ASCII text file. By default SET MEMOSOFTCR is ON.

Example

```
set memosofter on
memoread("text.txt", mfield)
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MEMOWIDTH

Class

Memos

Purpose

Adjust the display width of memo fields

Syntax

SET MEMOWIDTH TO <expN>

See Also

SET MEMOFORMAT, SET MEMOWINDOW, SET WP, MEMOREAD(), MEMOWRITE(),
MEMLINES(), MLINE(), MEMOEDIT(), TEXTEDIT(), MLCOUNT(), MEMOLINE()

Description

The SET MEMOWIDTH TO <expN> command adjusts the display width of a MEMO. This display width will be used as the default in reports. The SET MEMOWIDTH command also affects the MEMMLINES() and MLINE() functions, and the ? and ?? commands. The SET MEMOWIDTH TO command does not affect the memo-editing window. By default, MEMOWIDTH is 40.

Example

set memowidth to 60

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET MEMOWINDOW

Class

Screen Environment

Purpose

Edit memo fields in a pop-up window

Syntax

SET MEMOWINDOW AT DEFAULT | <expN1>,<expN2> TO <expN3>,<expN4>
SET MEMOWINDOW ON | OFF | (<expL>)

See Also

SET MEMOWIDTH, MEMOREAD(), MEMOWRITE(), MEMLINES(), MLINE(), MEMOEDIT(),
TEXTEDIT(), MLCOUNT(), MEMOLINE(), SET WP

Description

If MEMOWINDOW is ON, then memos can be edited in a 'pop-up' window in BROWSE, CHANGE, EDIT, and APPEND. The memo-editing window is activated at top row <expN1>, column <expN2> to bottom row <expN3>, column <expN4> by pressing the [EDIT FIELD] key when the cursor is placed on the memo field. The memo editor used with the window is the same as is used with the MEMOEDIT() and TEXTEDIT() functions. If MEMOWINDOW is set OFF, then the designated text editor that was set with the SET WP command will be opened. By default, MEMOWINDOW is ON and its coordinates are 5, 18 to 18, 62.

Example

set memowindow on
set memowindow at 10,10 to 20,60

Products

Recital Terminal Developer

SET MENU

Class

Screen Environment

Purpose

Determines whether a function key menu is displayed with full screen forms commands

Syntax

SET MENU ON | OFF | (<expL>)

See Also

SET HELP, APPEND, CHANGE, EDIT, INSERT, BROWSE, QUERY

Description

If SET MENU is ON, a function/control key menu is displayed at the top of the screen during full screen forms based commands such as APPEND, CHANGE, EDIT, INSERT, BROWSE and QUERY. In a system generated form, pressing the [TAB] key toggles the menu on and off. If a user defined form has been selected with the SET FORMAT TO command, no menu is displayed initially, however the [TAB] key toggles the display of a popup key help menu. By default, MENU is ON.

Example

```
set menu off
use patrons index events, dates, names
set filter to event = "BALLET"
browse
```

Products

Recital Terminal Developer

SET MENUBAR

Class

Screen Forms

Purpose

Enable or disable the menu bar in forms

Syntax

SET MENUBAR ON | OFF | (<expL>)

See Also

APPEND, EDIT, QUERY, BROWSE, SET PRERECORD, SET PREFORM, CREATE SCREEN, MODIFY SCREEN, READMODE()

Description

When the MENUBAR is set ON, it enables use of the [MENU] key from within forms to activate user-defined menus. If MENUBAR is set OFF, the [MENU] is disabled from inside forms for activating user-defined menus. This is particularly useful when using the same form for APPEND and EDIT operations. The default setting for MENUBAR is ON.

Example

```
if menuitem()="APPEND"
    set menubar off
    append
else
    set menubar on
    edit
endif
```

Products

Recital Terminal Developer

SET MESSAGE

Class

Screen Environment

Purpose

Control the display of messages

Syntax

```
SET MESSAGE TO <expC> | [<expN> [CENTER] [CENTRE]]  
SET MESSAGE AT <expN1>,<expN2> TO <expN3>,<expN4> | DEFAULT  
SET MESSAGE ON | OFF | (<expL>)
```

See Also

DIALOG BOX, MESSAGE, SET STATUS, SET SCOREBOARD

Description

The SET MESSAGE command is used to create a message, consisting of the character string <expC>, and to determine where and when it will be displayed. The MESSAGE TO <expC> command displays messages without waiting for any user input. This command is very useful for displaying helpful messages, during long operations, to let the user know what is happening. If the command STATUS is set ON, the <expC> is centered in the message line.

If SET MESSAGE TO is issued with no <expC> specified, the relevant message line will be cleared. If SET STATUS is OFF, then no message will be displayed. By default, STATUS is ON.

The SET MESSAGE command, with the AT <expN1> TO <expN2> option, can be used to position the message line at any chosen position on the screen. This is particularly useful when working with menus. SET MESSAGE AT DEFAULT reselects the default position setting of line 24, for all messages. SET MESSAGE can be optionally centered with the keywords CENTER or CENTRE.

If STATUS is OFF and SCOREBOARD is ON, messages appear on the top line of the screen. If STATUS is OFF and SCOREBOARD is OFF then messages are disabled unless MESSAGE is ON. Note that if messages are fully disabled (STATUS OFF, SCOREBOARD OFF and MESSAGE OFF), the [FIND] and [FINDNEXT] keys are ignored. By default, MESSAGE is ON.

Example

```
set status off  
set message at 10,10 to 10,50  
set message to "Name and address not; specified."
```

```
// Another example  
use patrons index events, dates, names  
seek "BALLET"  
if not found()  
    set message to "Key not found."  
endif
```

Products

Recital Mirage Server, Recital Terminal Developer

SET MOUSE

Class

Screen Forms

Purpose

Enable or disable free cursor movement on forms.

Syntax

SET MOUSE ON | OFF | (<expL>)

See Also

SET NAVIGATE, APPEND, CHANGE, CREATE SCREEN, EDIT, QUERY, READ

Description

When SET MOUSE is ON the cursor may be moved to any screen position on a form by using the cursor keys.

If SET MOUSE is ON and SET NAVIGATE is ON, pressing the [RETURN] key moves the cursor to the nearest field or menu item. When SET MOUSE is OFF the cursor is restricted to movement in the defined @...GET statements of the current screen format. By default SET MOUSE is OFF.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned MOUSE is set ON. If a value of .F. is returned MOUSE is set OFF.

Example

```
set mouse on
set navigate on
use demo
edit
```

Products

Recital Terminal Developer

SET MULTIUSER

Class

Environment

Purpose

Enable or disable all multi-user commands and functions

Syntax

SET MULTIUSER ON | OFF | (<expL>)

See Also

SET EXCLUSIVE, USE, UNLOCK, RLOCK, FLOCK, LOCKF (), LOCKR ()

Description

If MULTIUSER is set OFF, all multi-user commands and functions are disabled. As a result, all files are opened exclusively. The FLOCK() and RLOCK() functions will always return a .T.. Commands such as UNLOCK, SET EXCLUSIVE, and USE...EXCLUSIVE are ignored. This allows multi-user applications to be distributed and run with single-user licenses. By default, MULTIUSER is OFF.

Example

```
set multiuser off
use demo
edit
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET NAVIGATE

Class

Screen Forms

Purpose

Enable or disable random movement to fields

Syntax

SET NAVIGATE ON | OFF| (<expL>)

See Also

USE, APPEND, CHANGE, CREATE SCREEN, EDIT, QUERY, READ

Description

The SET NAVIGATE command controls cursor movement within a form. When SET NAVIGATE is ON, the cursor keys may be used to navigate within a form independent to the order defined by the @...GET statements which make up that form. When you are in a form that has been activated with the READ, EDIT, CHANGE, QUERY, or APPEND command, and SET NAVIGATE is ON, the cursor keys (UP, DOWN, LEFT, RIGHT) may be used to moved from field to field in any order rather than in order of the @...GET statements in the form. When SET NAVIGATE is OFF, the cursor moves sequentially from field to field, following the order of the activated @...GET statements.

Example

set navigate on

Products

Recital Terminal Developer

SET NEAR

Class

Indexing

Purpose

Sets 'soft' seeking of data

Syntax

SET NEAR ON | OFF | (<expL>)

See Also

SET SOFTSEEK, SEEK, FIND, INDEX, SOUNDEX()

Description

When SET NEAR is ON, searches of the table go to the record immediately following the potential location of the search key in the relevant file, if the key itself is not found. SET NEAR is synonymous with SET SOFTSEEK. Both SET NEAR and SET SOFTSEEK function when any valid expression is used with the SEEK command. It has no effect on relationships. By default, NEAR is OFF.

Example

```
set near on
use demo
select state
seek "MB"
? found()
```

.F.

```
? eof()
```

.F.

```
? state
```

MD

```
set near off
```

```
seek "MB"
```

```
? found()
```

.F.

```
? eof()
```

.T.

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET NOTIFY

Class

Recital Terminal Developer Environment

Purpose

Enable or disable the display of certain system messages

Syntax

SET NOTIFY ON | OFF | (<expL>)

See Also

SET MESSAGE, WAIT

Description

The SET NOTIFY allows you to disable some of the systems messages. When NOTIFY is set ON then certain systems messages are displayed, if OFF the display of certain system messages is suppressed.

Example

```
set notify off
```

Products

Recital Terminal Developer

SET NULL

Class

SQL Applications

Purpose

To determine NULL value support

Syntax

SET NULL ON | OFF

See Also

ALTER TABLE, CREATE TABLE, INSERT, EMPTY(), ISBLANK(), ISNULL(), SET NULLDISPLAY

Description

The SET NULL ON | OFF command is used to determine whether columns in a table support NULL values. With SET NULL ON, table columns will support NULL values by default. INSERT will insert a NULL into any column that does not have a value specified. With SET NULL off, NULL values are not supported by default. INSERT will insert a NULL into any column that does not have a value specified.. This default can be overridden by specifying the NULL or NOT NULL column constraint on an individual column.

SET NULL is OFF by default.

Example

```
set sqldialect to vfp
set null on
CREATE TABLE nullon (firstname c(20), lastname c(20))
INSERT INTO nullon (lastname) VALUES ("Smith")
? [SET NULL ON]
? [ISNULL() ], isnull(firstname)
? [EMPTY() ], empty(firstname)
wait
SET NULL ON
ISNULL() .T.
EMPTY() .F.
Press any key to continue...
```

```
set null off
CREATE TABLE nulloff (firstname c(20), lastname c(20))
INSERT INTO nulloff (lastname) VALUES ("Smith")
? [SET NULL OFF]
? [ISNULL() ], isnull(firstname)
? [EMPTY() ], empty(firstname)
wait
SET NULL OFF
ISNULL() .F.
EMPTY() .T.
Press any key to continue...
```

```
set null off
CREATE TABLE nulloff2 (firstname c(20) NULL, lastname c(20))
INSERT INTO nulloff2 (firstname,lastname) VALUES (NULL,"Smith")
? [SET NULL OFF, NULL Column Constraint]
? [ISNULL() ], isnull(firstname)
? [EMPTY() ], empty(firstname)
wait
    SET NULL OFF, NULL Column Constraint
    ISNULL() .T.
    EMPTY() .F.
    Press any key to continue...
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET NULLDISPLAY

Class

SQL Applications

Purpose

To specify the text displayed for NULL values

Syntax

SET NULLDISPLAY TO [<expC>]

See Also

ALTER TABLE, CREATE TABLE, INSERT, EMPTY(), ISBLANK(), ISNULL(), SET NULL

Description

The SET NULLDISPLAY command is used to specify the text displayed for NULL values. By default NULL values are displayed as .NULL.. The optional <expC> is used to specify alternative display text. If <expC> is omitted, then the display is reset to the default.

Example

```
set sqldialect to vfp
set null on
set heading off
CREATE TABLE nullon (firstname c(20), lastname c(20))
INSERT INTO nullon (lastname) VALUES ("Smith")
list off
      .NULL.      Smith
```

```
set nulldisplay to "<null>"
list off
      <null>      Smith
```

```
set nulldisplay to
list off
      .NULL.      Smith
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ODOMETER

Class

Recital Terminal Developer Environment

Purpose

Determine how often the counter is updated in commands that display their progress

Syntax

```
SET ODOMETER AT <expN1>, <expN2> | DEFAULT  
SET ODOMETER ON | OFF | (<expL>)  
SET ODOMETER TO REVERSE | BOLD | NONE | <expN>
```

See Also

SET TALK, SET PERFMETER

Description

When SET TALK is ON, certain commands display their progress on the screen. Commands such as INDEX, PACK, SORT etc. ‘count’ on the screen to inform you of their progress. The SET ODOMETER TO <expN> command allows you to specify the units in which the counting should be performed. By default, the counting is performed in units of one. When SET TALK is OFF, SET ODOMETER TO will not operate unless SET ODOMETER is ON.

You may position the counter display using AT <expN1>, <expN2> which specify the row and column coordinates, or place it back to the default position using the DEFAULT keyword. The BOLD, REVERSE, and NONE keywords allow you to control the display attributes of the counter.

Example

```
set talk off  
set odometer on  
set odometer to 10  
set odometer at 10,43  
set message to “Pack in progress”  
pack  
set odometer off
```

Products

Recital Terminal Developer

SET ONERROR

Class

Error Handling and debugging

Purpose

Disable ON ERROR error handling

Syntax

SET ODOMETER ON | OFF | (<expL>)

See Also

ON ERROR, SET ERRORVERSION

Description

When SET ONERROR is OFF, any ON ERROR error handling settings are disabled. To reinstate ON ERROR error handling, SET ONERROR must be set ON and the ON ERROR <command> must be reissued. This is particularly useful for disabling ON ERROR error handling on an application or system-wide basis to allow use of the default error.mem error trapping.

Example

on error dialog box [Error]

? on("error")

dialog box [Error]

set onerror off

? on("error")

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET OPTIMIZE

Class

Information Center

Purpose

Enable or disable the query optimization

Syntax

SET OPTIMIZE ON | OFF | (<expL>)

See Also

INDEX, REINDEX, SET INDEX, SET ORDER, SET VIEW, USE, SET CATALOG, SET DESIGN, INFO, DESIGN

Description

The SET OPTIMIZE command enables or disables the query optimizer in the Recital Information Center. When SET OPTIMIZE is ON, an index order which optimizes the current FOR condition is chosen. Based on the requirements specified in the FOR condition, the best-suited index among those associated with the currently active table is selected. If no index is found which can optimize the query, a sequential search is performed. When SET OPTIMIZE is OFF, a sequential search will always be performed in connection with a FOR condition. The SET INDEX and USE commands can both open a list of single indexes to be associated with the active table. Up to a maximum of 20 single indexes may be opened per table. Production indexes are opened automatically when a table is opened and can have a maximum of 128 tag orders defined.

The query optimizer operates in the Recital Information Center. The Information Center is a powerful work surface from which any type of data may be accessed, viewed, modified, and organized. With the Information Center you can organize files into catalogs which represent a single application, an application interface, or development project. Depending on your orientation to the data, the Information Center may serve as an application interface, and an organizing and developing tool.

Example

```
set optimize on
set catalog to sales
info
```

Products

Recital Terminal Developer

SET OPTLOG

Class

Environment

Purpose

To specify whether logical expression optimization should take place

Syntax

SET OPTLOG ON | OFF

See Also

DB_OPTLOG

Description

The SET OPTLOG command is used to specify whether logical expression optimization should take place or not. In post-8.2 versions, the SET OPTLOG command can be used to configure the optimization of logical expressions containing the OR operator as well as those with the AND operator.

Example

```
// No error, even if crash() function does not exist
set optlog on
? .F. and crash()
// Gives error if crash() function does not exist
set optlog off
? .F. and crash()
```

```
// No error, even if crash() function does not exist
set optlog on
? .T. or crash()
// Gives error if crash() function does not exist
set optlog off
? .T. or crash()
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET ORDER

Class

Indexing

Purpose

Select the master index file from the active index file list

Syntax

SET ORDER TO <expN> | [TAG] <tag-name> [OF <.dbx filename>] | <single index filename>
[ASCENDING | DESCENDING]

See Also

SET INDEX, USE, INDEX, NDX(), INDEXKEY(), INDEXORDER(), MDX(), TAG(), TAGNO(), TAGCOUNT()

Description

The SET ORDER TO <expN> command selects an open index file as the master index. The master index is the index file searched during the FIND and SEEK commands, and also defines the order in which records are retrieved from the table. The master index may be selected with a number, <expN>. This is the sequential number of the index in the current list of indexes associated with a .dbf file. Single index files (.ndx/.idx) may also be selected by name, <single index filename>. If .dbx files are in use, the master index may be selected using a tag name. If there are tags that have the same name, they may be specified exactly with the OF <.dbx filename> qualifier.

The SET ORDER TO 0 or SET ORDER TO commands cause index files to be ignored when retrieving records, but they will still be updated if any changes are made to the table. The SET ORDER TO command is only valid with tables that have index files associated with them.

ASCENDING | DESCENDING

The optional ASCENDING | DESCENDING can be used to determine the order in which records are accessed and displayed.

Example

use patrons index address.dbx
set order to tag zip of address

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PAGELENGTH

Class

Printing

Purpose

Governs pagination in output files

Syntax

SET PAGELENGTH TO <expN>

See Also

SET HEADING TO, SET PAGEWIDTH TO, TYPE, LIST, DISPLAY, REPORT, BROWSE, SET DESCRIPTION

Description

The SET PAGELENGTH TO <expN> command sets the length of pagination in an output file created from one of the DISPLAY | LIST...TO FILE or TYPE FILE commands. When used in conjunction with the LIST | DISPLAY ... TO FILE, SET DESCRIPTION, SET PAGEWIDTH and SET HEADING TO commands, this provides the ability to produce quick reports.

Example

```
set pagelength to 60
set heading to single
use demo
list fields acc_prefix, acc_no, ord_value, name;
    to file accounts.txt;
    for ord_date =(“01/01/00”)
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PAGEWIDTH

Class

Printing

Purpose

Defines page width in output files

Syntax

SET PAGEWIDTH TO <expN>

See Also

SET HEADING, LIST, DISPLAY, REPORT, BROWSE, SET DESCRIPTION

Description

The SET PAGEWIDTH TO <expN> command sets the width in an output file created from one of the DISPLAY | LIST...TO FILE commands. The page number is centered accordingly on the bottom line of the page. When used in conjunction with the LIST | DISPLAY ... TO FILE, SET DESCRIPTION, SET PAGELength and SET HEADING TO commands, SET PAGEWIDTH provides the ability to produce quick reports.

Example

```
set pagelength 60
set pagewidth to 65
use demo
list fields acc_prefix, acc_no, ord_value, name;
    to file accounts.txt;
    for ord_date = ("01/01/00")
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PATH

Class

Environment

Purpose

Specify the directory search path

Syntax

SET PATH TO <path-list>

See Also

SET DEFAULT, DIR, FILE()

Description

The SET PATH TO <path-list> command specifies the directory search path used to locate a file that is not found in the current directory. Each of the directory names specified in the list should be separated with a ',' or a ';'. Up to 10 different directory paths can be specified. Issuing the SET PATH TO command without a <path-list> removes the current search path if one is present. If no SET PATH TO <path-list> command has been issued, then files will only be searched for in the current directory.

It is important to note that when trying to open a table, a search is made in each directory in the path list in turn, attempting to open the table with read/write access. If the table cannot be opened with read/write access after trying all directories in the search path, the process is then repeated, attempting to open the table for read only access. If read only access is granted, the table is marked as read only and no update operations will be allowed on it. This allows standard operating system file protection to be enforced on tables. The DIR command lists all files in each of the directories specified in the path list.

Example

```
set path to c:\sales\
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PAUSE

Class

Recital Terminal Developer Environment

Purpose

Enable or disable screen pauses for the Recital/SQL SELECT statement

Syntax

SET PAUSE ON | OFF | (<expL>)

See Also

DISPLAY, LIST, SET SQL

Description

The SET PAUSE ON command causes the Recital/SQL SELECT command to pause after every screen of output until a key is pressed to continue, or the [ABANDON] key is pressed to cancel. When SET PAUSE is off, records retrieved with the Recital/SQL SELECT command scroll continuously until the last record is displayed.

Example

```
use accounts
set pause on
set sql on
select * from accounts;
```

Products

Recital Terminal Developer

SET PCACHE

Class

Performance and Optimization

Purpose

Read programs completely into memory before execution commences

Syntax

SET PCACHE ON | OFF | (<expL>)

See Also

SET CACHELOAD, SET GCACHE, SET ICACHE, SET DCACHE

Description

If SET PCACHE is ON, then programs executed with the DO command are read completely into memory before execution commences. Screen format files activated with the SET FORMAT TO command are also read completely into memory. By default, PCACHE is ON.

Example

```
set pcache on
do accounts
use patrons index events, dates, names
set format to dataentry
change
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PCEDIT

Class

Xbase Compatibility

Purpose

Add Xbase compatibility for cursor movement in forms.

Syntax

SET PCEDIT ON | OFF | (<expL>)

See Also

APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCEDIT command manages cursor movement on forms. By default PCEDIT is OFF, which uses native Recital cursor control within forms. With PCEDIT ON the cursor control behaves like the PC Xbase languages. Changes in behavior while editing forms with PCEDIT set ON include:

- Numeric fields line up on decimal point when the decimal is entered.
- Cursor LEFT at the start of a field moves to the last cursor position on the previous field.
- Cursor RIGHT at the start of a field moves into the field in OVERTYPE editing mode.
- Control keys can be used at any point in the field to process the form. This allows [EXIT/SAVE] or [ABANDON] to be entered while in the middle of a field in order to exit the form.
- @...PROMPT menu processing now forces the selection to the top of the list when pressing [HOME] or to the end of the list when pressing [END] with PCEDIT ON.

Example

```
set pcedit on
use customer
set format to customer
edit
```

Products

Recital Terminal Developer

SET PCEXACT

Class

Xbase Compatibility

Purpose

Determines how the SET EXACT command compares strings

Syntax

SET PCEXACT ON | OFF | (<expL>)

See Also

EDITFIELD(), FILETYPE(), INDEXEXT(), LEFT(), LPAD(), LTRIM(), RPAD(), RIGHT(), RTRIM(), SOUNDEX(), SUBSTR(), TRIM(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET EXACT, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCEXACT command affects the way that the SET EXACT command performs string comparisons. If SET PCEXACT is ON, and SET EXACT is ON, the strings compared need not be an exact match in length. PCEXACT allows for the presence of spaces on the right side of strings. If PCEXACT is ON, and SET EXACT is OFF, PCEXACT will have no affect on string comparisons. If SET PCEXACT is OFF, and SET EXACT is ON, strings are compared for exact length and content. If SET PCEXACT is OFF, and SET EXACT is OFF, strings are compared up to the length of the shortest string. By default, PCEXACT is OFF.

Example

```
set exact on
? "abc" = "abc"    "
.F.
set pcexact on
? "abc" = "abc"    "
.T.
set pcexact off
set exact off
? "abc" = "abc"    "
.F.
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PCFILTER

Class

Xbase Compatibility

Purpose

Provide compatibility with dBASE filters

Syntax

SET PCFILTER ON | OFF | (<expL>)

See Also

EDITFIELD(), FILETYPE(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FILTER, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCLOCKING, SET PCKEYS, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The PCFILTER command allows compatibility with dBASE filter conditions.

When PCFILTER is ON, records that are outside the filter condition can be accessed and updated in the same way as in dBASE. When PCFILTER is set OFF, records that are outside the filter condition are never accessible. This command has been added to aid in the porting of applications that use this feature in dBASE. The default for PCFILTER is OFF.

Example

```
set pfilter on
goto 23      && record outside of filter
replace fld_name with "new value"
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PCFKEYS

Class

Keyboard Events

Purpose

Enable or disable ON KEY LABEL Ctrl-<letter> functionality.

Syntax

SET PCFKEYS ON | OFF | (<expL>)

See Also

ON KEY, SET PCKEYS, SET KEY TO, SET KEY...TO

Description

The SET PCFKEYS ON command enables or disables the use of Ctrl-<letter> keynames by the ON KEY LABEL command. ON KEY LABEL keynames are as follows:

Key	Keyname
F1 to F10	F1, F2, F3 ...
CURSOR LEFT	Leftarrow
CURSOR RIGHT	Rightarrow
CURSOR UP	Uparrow
CURSOR DOWN	Downarrow
PAGE UP	PgUp
PAGE DOWN	PgDn
DELETE	Del
INSERT	Ins
TAB	Tab
CTRL A TO CTRL Z	Ctrl-A, Ctrl-B, Ctrl-C

By default SET PCFKEYS is OFF.

Example

```
set pcfkeys on
on key label Ctrl-K do k_proc
```

Products

Recital Mirage Server, Recital Terminal Developer

SET PCGRAPHICS

Class

Xbase Compatibility

Purpose

Map PC line drawing graphics characters.

Syntax

SET PCGRAPHICS ON | OFF | (<expL>)

See Also

APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, WAIT, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), INKEY(), LASTKEY(), READKEY(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCLOCKING, SET PCKEYS, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

If SET PCGRAPHICS is ON, then IBM-PC line drawing graphics characters are mapped into the appropriate character sequence for the issuing terminal.

This command assists the portability of dBASE programs that use the line drawing characters of the IBM-PC, e.g., vertical lines, horizontal lines, box corners etc.

This command allows for the optional logical expression <expL> to be evaluated. If a value of .T. is returned, PCGRAPHICS is set ON. If a value of .F. is returned, PCGRAPHICS is set OFF. By default PCGRAPHICS is OFF.

Example

```
set pcgraphics on
use customer
set format to customer
edit
```

Products

Recital Terminal Developer

SET PCKEYS

Class

Xbase Compatibility

Purpose

Translate function key codes to PC equivalents.

Syntax

SET PCKEYS ON | OFF | (<expL>)

See Also

APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, WAIT, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), INKEY(), LASTKEY(), READKEY(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCKEYS ON command causes the INKEY(), READKEY() and LASTKEY() functions, and the WAIT command, to translate the codes for function keys pressed at the keyboard to their IBM-PC counterparts. A table summarizing the codes is shown below.

Keypad List	PC Key	Wait	Inkey()	Readkey()	Lastkey()
[1]	F1	0	28	36	28
[2]	F2	255	-1		-1
[3]	F3	254	-2		-2
[4]	F4	253	-3		-3
[5]	F5	253	-4		-4
[6]	F6	251	-5		-5
[7]	F7	250	-6		-6
[8]	F8	249	-7		-7
[9]	F9	248	-8		-8
[0]	F10	247	-9		-9
[CURSOR LEFT]	CURSOR LEFT	19	19	0	19
[CURSOR RIGHT]	CURSOR RIGHT	4	4	1	4
[CURSOR UP]	CURSOR UP	5	5	4	5
[CURSOR DOWN]	CURSOR DOWN	24	24	5	24
[PAGE UP]	PgUp	18	18	6	18
[PAGE DOWN]	PgDn	3	3	7	3
[ABANDON]	Esc	27	27	12	27
[EXIT/SAVE]	End	23	23	14	23
[PAN LEFT]	Ctrl <-	26	26	8	26
[PAN RIGHT]	Ctrl ->	2	2	9	2

When PCKEYS is ON the '/' key cannot be used to activate the menu bar. The [MENUBAR] key must be used instead. This allows the '/' to be input into a field on a form.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned PCKEYS is set ON. If a value of .F. is returned PCKEYS is set OFF. By default SET PCKEYS is OFF.

Example

set pckey on

? inkey(0)

// Press key to obtain value

Products

Recital Mirage Server, Recital Terminal Developer

SET PCLOCKING

Class

Xbase Compatibility

Purpose

Determine whether multiple locks may be applied

Syntax

SET PCLOCKING ON | OFF | (<expL>)

See Also

APPEND, CHANGE, EDIT, UNLOCK, USE, ACCESS(), EDITFIELD(), FILETYPE(), FLOCK(), INDEXEXT(), LOCK(), LOCKF, LOCKR, RLOCK(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET EXCLUSIVE, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET LOCK, SET LOCKWAIT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCLOCKING command determines whether multiple locks may be applied to records or the currently active table. If PCLOCKING is set ON, previous locks will be canceled as FLOCK() or RLOCK() functions attempt to lock records or files that are all ready locked. The previous lock will be removed regardless of the success or failure of the FLOCK() or RLOCK() function. When PCLOCKING is set OFF, multiple applications of the FLOCK() and RLOCK() functions are allowed. Please note that file and record locking is performed automatically. In most situations, the SET PCLOCKING command and the RLOCK() and FLOCK() functions are unnecessary, and are included for compatibility with programs written in other products.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, PCLOCKING is set ON. If a value of .F. is returned PCLOCKING is set OFF. Note that the <expL> should be enclosed in round brackets. By default, PCLOCKING is OFF.

Example

```
set pclocking on
do while not rlock()
    set message to "Record in use."
    sleep 2
enddo
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PCPICTURE

Class

Xbase Compatibility

Purpose

Truncate extra decimal places for dBASE compatibility

Syntax

SET PCPICTURE ON | OFF | (<expL>)

See Also

@...SAY, APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, CEILING(), EDITFIELD(), FILETYPE(), FLOOR(), FMT(), INDEXEXT(), ROUND(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCSAYS, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCPICTURE command allows for dBASE compatibility with the handling of the decimal portion of numeric data in picture statements. When PCPICTURE is set ON, if a picture statement has fewer decimal places than the actual variable or field, the extra decimal places are truncated. If PCPICTURE is OFF, then the numeric data is properly rounded. This command has been added to aid in the porting of applications that use this feature. By default, PCPICTURE is OFF.

Example

```
num = 34.675
set pcpicture on
@ 0,0 say num picture 99.9
    34.6
set pcpicture off
@ 0,0 say num picture 99.9
    34.7
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PCSAYS

Class

Xbase Compatibility

Purpose

Update @...SAYs in format files for dBASE compatibility

Syntax

SET PCSAYS ON | OFF | (<expL>)

See Also

@...SAY, APPEND, CHANGE, CREATE SCREEN, EDIT, MODIFY SCREEN, EDITFIELD(), FILETYPE(), FMT(), INDEXEXT(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET FORMAT, SET INDEXEXT, SET MEMOEXT, SET PCEDIT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCUNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The @...SAY commands which reference field values in a format file are not usually updated as you page up and down a table while editing. This occurs by design to stop unnecessary terminal output overhead. The SET PCSAYS command allows for dBASE compatibility with updating @...SAYs when paging through the records in a format file. When PCSAYS is set ON, @...SAY commands are redisplayed on the screen while paging up and down the table. If PCSAYS is OFF, then @...SAY commands are not updated. This command has been added to aid in the porting of applications that use this feature. By default, PCSAYS is OFF.

Example

```
set pcsays on
edit
```

Products

Recital Terminal Developer

SET PCUNIQUE

Class

Xbase Compatibility

Purpose

Cause 'unique' indexes to be treated as they are in dBASE

Syntax

SET PCUNIQUE ON | OFF | (<expL>)

See Also

APPEND, INDEX, EDITFIELD(), FILETYPE(), INDEXEXT(), UNIQUE(), SET CLIPPER, SET CLIPPER5, SET COMPATIBLE, SET EDITFIELD, SET FILECASE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET PCEXACT, SET PCFILTER, SET PCGRAPHICS, SET PCKEYS, SET PCLOCKING, SET PCPICTURE, SET PCSAYS, SET PCUNIQUE, SET UNIQUE, DB_FOXPLUSBUGS, DB_FOXPROKEYS, DB_SAMBA

Description

The SET PCUNIQUE command causes unique indexes to be treated in the same manner as dBASE. When PCUNIQUE is set ON, it 'hides' records with the same key value in the UNIQUE index file. Duplicate records can be added to the UNIQUE index file but will not be displayed. However, the GOTO command will still position the record pointer on the record. When SET PCUNIQUE is OFF, duplicate records cannot be entered into the UNIQUE index file. If the attempt is made from the FORMS system, an error message is displayed, and a new index key can be entered, or the [ABANDON] key pressed. If the attempt is made elsewhere, then an error is returned. The default for PCUNIQUE is OFF.

Example

```
set pcunique on
append from temp
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PERFDIAL

Class

Recital Terminal Developer Environment

Purpose

Control performance dial display

Syntax

SET PERFDIAL ON | OFF | (<expL>)

See Also

SET TALK ON, SET ODOMETER, SET PERFMETER

Description

The SET PERFDIAL command controls the display of a performance dial within the performance meter. The PERFMETER command controls the display of the performance meter. The performance meter charts your progress through commands that would normally count on the screen. The performance dial spins according to the speed at which I/O operations are being performed. When PERFMETER and PERFDIAL are set ON, a box is displayed on the screen containing information on which command is executing and what percentage of the operation is complete. The performance dial displays in the upper left corner of the performance meter. The performance meter displays for the following commands:

- AVERAGE
- COUNT
- COMPILE
- COPY
- INDEX
- PACK
- REINDEX
- SET COMPILE ON
- SUM
- TOTAL

If SET TALK is ON and SET PERFMETER is OFF progress for these commands is displayed as a record counter.

Example

```
set perfmeter on
set perfdial on
reindex all
```

Products

Recital Terminal Developer

SET PERFMETER

Class

Recital Terminal Developer Environment

Purpose

Control performance meter display

Syntax

SET PERFMETER ON | OFF | (<expL>)

SET PERFMETER [AT <expN>]

See Also

SET TALK ON, SET ODOMETER, SET PERFDIAL

Description

The SET PERFMETER command controls the display of a performance meter. The performance meter charts your progress through commands that would normally count on the screen. When PERFMETER is set ON, a box is displayed on the screen containing information on which command is executing and what percentage of the operation is complete.

The optional AT <expN> qualifier repositions the PERFMETER to the row specified by <expN>.

Example

```
set perfmeter on
reindex all
```

Products

Recital Terminal Developer

SET POINT

Class

Numeric Data

Purpose

Change the character used as a decimal point

Syntax

SET POINT TO [<expC>]

See Also

SET DECIMALS, SET SEPARATOR

Description

The SET POINT TO command is used to change the decimal point character from a period (.) to the character specified by <expC>. The decimal point character is changed for output purposes only. Regardless of the character specified by the SET POINT TO command, the period is used when a decimal value is stored in a field or a memory variable. When entering a decimal value into a field, <expC> may be used as the decimal point, but the character will be changed to a period (.) when the record is saved. If the character expression, <expC>, evaluates to more than one character, the SET POINT TO command only uses the first character in the string. Used without the optional <expC>, the SET POINT command sets the decimal point back to a period (.).

Example

set separator to “.”

set point to “,”

display all ord_value, customer

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET POSTFORM

Class

Screen Forms

Purpose

Define a postform trigger to execute as the format file is exited

Syntax

SET POSTFORM TO [<procedure-name> | (<expC>)]

See Also

SET PREFORM, SET PRERECORD, SET POSTRECORD, @...GET, CREATE SCREEN, MODIFY SCREEN, CREATE, MODIFY STRUCTURE, SET JOURNAL

Description

The POSTFORM trigger procedure executes a procedure upon exiting a format file with the [EXIT/SAVE] key. The SET POSTFORM TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The SET POSTFORM TO command clears the active <procedure-name>.

The SET POSTFORM trigger can also be defined from within the SCREEN PAINTER work surface. The POSTFORM procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER.

Example

```
set postform to save_count
edit
```

Products

Recital Mirage Server, Recital Terminal Developer

SET POSTMENU

Class

Screen Forms

Purpose

Designate an event-driven trigger to execute as a menu in a form file is exited

Syntax

SET POSTMENU TO [<program | procedure> | (<expC>)]

See Also

@...GET, @...MENU, CREATE SCREEN, MODIFY SCREEN, SAVE MENU, SET PREMENU, SET PREFORM, SET POSTFORM, SET PRERECORD, RESTORE MENU

Description

The POSTMENU trigger procedure executes a procedure whenever a menu is exited from within the format file. The SET POSTMENU TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The SET POSTMENU trigger can also be defined from within the SCREEN PAINTER work surface. The POSTMENU procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER. The SET POSTMENU TO command clears the active <program/procedure>.

Example

```
procedure post_proc
use employee
set format to employee
return

clear
clear all
set exclusive off

use company
set form to company
set postmenu to post_proc
edit
return
```

Products

Recital Mirage Server, Recital Terminal Developer

SET POSTRECORD

Class

Screen Forms

Purpose

Designate an event-driven trigger to execute as a record in a form file is exited

Syntax

SET POSTRECORD TO [<procedure-name> | (<expC>)]

See Also

SET PREFORM, SET POSTFORM, SET PRERECORD, @...GET, CREATE SCREEN, MODIFY SCREEN

Description

The POSTRECORD trigger procedure executes a procedure whenever a record is updated from within the format file. The SET POSTRECORD TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The commands in the POSTRECORD trigger procedure are executed before the record is updated and unlocked. Conditional parsing of the data entered may be performed at this time and the resulting condition returned. If .F. is returned, the update is not performed and the user is placed back into the form. Only when .T. is returned from the POSTRECORD trigger is the record updated.

The [EXIT/SAVE] and [PAGE DOWN] keys will call the POSTRECORD trigger from EDIT, CHANGE, INSERT and APPEND. The [PAGE UP], [FIND] and [MENUBAR] keys will call it from EDIT, CHANGE and INSERT. POSTRECORD triggers are table specific, and will execute when records in the same table are updated. A form that contains records from different tables may employ POSTRECORD triggers from each represented table. Table fields, for example, may activate a different POSTRECORD trigger for each parent child relationship. The SET POSTRECORD trigger can also be defined from within the SCREEN PAINTER work surface. The POSTRECORD procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER.

If the POSTRECORD trigger is a <.prg filename> instead of a <procedure-name>, the format file can be used both interactively and from the Recital Information Center without needing any procedures to be predefined. The SET POSTRECORD TO command clears the active <procedure-name>. The REPLACE command cannot be used in a POSTRECORD trigger procedure to update records in the current view. The REPLACE() function is used for this purpose.

Example

```
procedure timestamp
replace(timefield, time())
return .T.
```

```
select 1
use payroll alias payroll
set form to formname
select 2
use audit_file alias audit_file
select payroll
set postrecord to timestamp
edit
```

Products

Recital Mirage Server, Recital Terminal Developer

SET PRECISION

Class

Xbase Compatibility

Purpose

Language compatibility only.

Syntax

SET PRECISION TO [<expN>]

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

set precision to 4

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PREFORM

Class

Screen Forms

Purpose

Preform trigger procedure to execute before a form is displayed

Syntax

SET PREFORM TO [<procedure-name> | (<expC>)]

See Also

SET POSTFORM, SET PRERECORD, SET POSTRECORD, @...GET, CREATE SCREEN, MODIFY SCREEN

Description

The PREFORM trigger procedure executes a procedure upon entering a form. The SET PREFORM TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The SET PREFORM TO command clears the active <procedure-name>.

The SET PREFORM trigger can also be defined from within the SCREEN PAINTER work surface. The PREFORM procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER.

Example

```
procedure intiform
// Display the contents on a memo
memosay(comp_hist,2,17,15,61)
return
```

```
select 1
use payroll alias payroll
set form to formname
set preform to initform
edit
```

Products

Recital Mirage Server, Recital Terminal Developer

SET PREMENU

Class

Screen Forms

Purpose

Designate an event-driven trigger procedure to execute prior to entering a menu

Syntax

SET PREMENU TO [<procedure-name> / (<expC>)]

See Also

@...GET, @...MENU, CREATE SCREEN, MODIFY SCREEN, SAVE MENU, SET PREFORM, SET POSTFORM, SET POSTMENU, SET POSTRECORD, RESTORE

Description

The PREMENU trigger procedure executes a procedure before the first @...MENU is activated from within a form. The SET PREMENU TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The PREMENU trigger is called when the [MENUBAR] is activated. The SET PREMENU TO command clears the active <procedure-name>.

The SET PREMENU trigger can also be defined from within the SCREEN PAINTER work surface. The PREMENU procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER.

Example

```
procedure pre_security
// Check group ID number, if not valid, prohibit menu access
if getgid() < 100
    dialog box "Restricted Menu Options"
    keyboard(chr(ctrl('g')))
endif
return
```

Products

Recital Mirage Server, Recital Terminal Developer

SET PRERECORD

Class

Screen Forms

Purpose

Designate an event-driven trigger procedure to execute prior to the first @...GET

Syntax

SET PRERECORD TO [<procedure-name> | (<expC>)]

See Also

SET PREFORM, SET POSTFORM, SET POSTRECORD, @...GET, CREATE SCREEN, MODIFY SCREEN

Description

The PRERECORD trigger procedure executes a procedure after the record is locked and the data has been read, but before the first @...GET is activated. The SET PRERECORD TO <procedure-name> command defines the procedure or program name to be executed. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. The PRERECORD trigger is called when the FORM is first activated with the EDIT, CHANGE, INSERT and APPEND commands. Once the form is active, the [PAGE UP], [PAGE DOWN] and [FIND] keys will call the PRERECORD trigger procedure. The SET PRERECORD TO command clears the active <procedure-name>.

PRERECORD triggers are table specific, and will execute before @...GETs in the same table are activated. A form which contains records from different tables may employ PRERECORD triggers from each represented table. Table fields, for example, may activate a different POSTRECORD trigger for each parent child relationship. The SET PRERECORD trigger can also be defined from within the SCREEN PAINTER work surface. The PRERECORD procedure option from the TRIGGERS menu is used to define the trigger name. The [HELP] key may be pressed here to edit the trigger file from inside the SCREEN PAINTER.

Example

```
procedure disp_memo
memosay(comp_hist,2,17,15,61)
return
```

```
select 1
use payroll alias payroll
set form to formname
set prerecord disp_memo
edit
```

Products

Recital Mirage Server, Recital Terminal Developer

SET PRINT

Class

Printing

Purpose

Direct output text to the printer

Syntax

SET PRINT ON | OFF | (<expL>)

See Also

SET PRINTER, SET DEVICE, @...SAY, EJECT, PRINT, SETPRC(), PROW(), PCOL(), PRINTSCREEN()

Description

If SET PRINT is ON, then any textual output which is sent to the screen is also sent to the printer. Full screen forms generated for commands such as APPEND, CHANGE, EDIT, INSERT, and QUERY are not sent to the printer. If you wish to send @...SAY output to the printer and use the PROW() and PCOL() functions, the command SET DEVICE TO PRINT should also be issued.

The SET PRINTER TO command can be used to redirect printer output to a printer attached to a serial communications line, or to cause the printer output to be spooled to the system printer. See the SET PRINTER command for full details. By default, PRINT is OFF.

Example

```
set print on
use patrons index events, dates, names
list all for event = "BALLET"
set print off
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PRINTER

Class

Printing

Purpose

Redirect printer output

Syntax

SET PRINTER TO [<device> | <filename> [ADDITIVE] | \\SPOOLER | <os-command> | <expC>]

See Also

SET PRINT, PRINT, SETPRC(), GETENV()

Description

The SET PRINTER TO <device> | <filename> command causes output to the printer to be redirected into a file or to a printer attached to a serial communications line. The <device> | <filename> is the name of the target device or filename. The environment variable DB_PRINT, which is defined in the product login file (login.com, profile.db or profile.uas), contains the command that is used to print a text file. This command can be set to a user specific value when the user logs into the system. You can therefore associate groups of people with specific printers, or change the options specified as default with the print command depending on the specific file to be printed.

ADDITIVE

If the printer output is being directed to a file, specifying the ADDITIVE keyword will cause the output to be appended to any existing file contents rather than overwriting the file.

The SET PRINTER TO \\SPOOLER command causes all textual output to the printer to be stored in a text file on the disk, and then queued to the printer queue for printing when the output is finished. Any commands which have the TO PRINT clause specified cause the current contents of the spool file to be sent to the system printer (as specified in DB_PRINT).

The SET PRINTER TO <os-command> | <expC> command works in the same way as SET PRINTER TO \\SPOOLER, except it prints the temporary file with the <os-command>. The <os-command> is any valid operating system print command. The <expC> must evaluate to a valid operating system print command. The SET PRINTER TO command on its own closes the current print file or device. If the SET PRINTER TO \\SPOOLER command had been specified, the text file is sent to the printer queue.

Example

```
set printer to \\spooler
set view to orders
set print on
list status
set print off
set printer to
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PROCEDURE

Class

Environment

Purpose

Open a procedure library file

Syntax

SET PROCEDURE TO [<.prg filename> | (<expC>)] [ADDITIVE]

See Also

CLOSE PROCEDURE, DO, FUNCTION, LINK, LIST PROCEDURE, PARAMETERS, PROCEDURE

Description

The SET PROCEDURE TO <.prg filename> command opens the specified procedure library file, scans the contents of it, and records the names and positions of the procedures defined within it. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. You can place as many procedures as you want in a procedure library file.

The settings of SET COMPILE and SET DEVELOPMENT (and Operating System permissions and disk space) will determine whether procedure library files are automatically compiled when the SET PROCEDURE TO command is issued.

If the optional ADDITIVE keyword is specified then any procedures that are already open are left open and the new procedure library is added. You can open up to 10 procedure files at any one time. The SET PROCEDURE TO command, without any <.prg filename> specified, closes all active procedure library files. A closed library file discards any knowledge of where the procedures within reside. The CLOSE PROCEDURE command provides the same functionality. Closing a procedure file by a procedure defined within it has undefined results.

Procedures are declared by the PROCEDURE or FUNCTION keyword followed by the procedure name. The RETURN statement denotes the end of a procedure. RETURN statements can be used within the procedure provided they are used within DO...WHILE, DO...CASE, or IF...ELSE...ENDIF statements. No syntax checking is carried out while the procedure library file is being scanned. The syntax checking is only carried out when the procedure is executed. Procedure library files are supported by Recital to provide compatibility with dBASE. However, you can declare procedures anywhere in Recital programs. A procedure declaration must have been encountered before the procedure is executed with a DO command. The active procedures and functions can be listed with the LIST PROCEDURE command.

Example

```
set procedure to statistics
do std with "patrons", "seats"
set procedure to
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET PROMPT

Class

Recital Terminal Developer Environment

Purpose

Specify the interactive prompt

Syntax

SET PROMPT TO <expC> | DEFAULT

See Also

SET SQLPROMPT, SET TIME

Description

The SET PROMPT TO <expC> command allows you to specify the interactive prompt. The prompt string can be a maximum of 10 characters long. The SET PROMPT TO DEFAULT command resets the prompt back to '>'. By default, the prompt is '>'.

Example

```
> set prompt to ". "  
. set prompt to default  
>
```

Products

Recital Terminal Developer

SET PSHARE

Class

Performance and Optimization

Purpose

Cause compiled programs and procedure libraries to load into shared memory

Syntax

SET PSHARE ON | OFF | (<expL>)

See Also

COMPILE, DO, LINK, SET COMPILE, SET DEVELOPMENT

Description

The SET PSHARE ON command causes compiled '.dbo' programs and procedure libraries to load into shared memory. When PSHARE is set ON, the same physical memory allocated to an executed program referenced when the program is loaded again. Conserving physical memory in this way allows for more users per system. UNIX allocates shared memory segments physically above the 'heap' (dynamic memory) storage area. Insufficient memory may result in the 'heap' storage area as the shared memory stack increases in size. To adjust the dynamic memory size, the environment variable DB_HEAPSIZE may be set to a specified allocation block in units of 1K. DB_HEAPSIZE is defined in the file profile.db, and defaults to 32K. The shared memory facility must be built into the UNIX kernel in order for SET PSHARE ON to work. Please note that on UNIX system V, the kernel parameter SHMMAX (maximum shared memory size) should be set to a value capable of holding your largest program. The kernel parameter SHMALL (total shared memory system wide) may also need adjusting. If SET PSHARE is ON, and there is insufficient shared memory available, memory is allocated privately.

Example

```
set pshare on
do bigjob
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET QUERYMODE

Class

Table Basics

Purpose

Enable or disable update toggle in full screen forms based commands

Syntax

SET QUERYMODE ON | OFF | (<expL>)

See Also

SET UPDATE

Description

If SET QUERYMODE OFF and SET UPDATE OFF are active then the [UPDATE MODE] key cannot be used to toggle into update mode. This prevents database updates from being performed through interactive forms based operations such as CHANGE, EDIT, and BROWSE. When QUERYMODE is OFF, the function keys that allow the user to position on other records through the CHANGE and EDIT forms are disabled. By default, QUERYMODE is ON.

Example

```
use patrons index events, dates, names
set querymode off
set update on
seek "BALLET"
edit
```

Products

Recital Terminal Developer

SET READEXIT

Class

Screen Forms

Purpose

Control of read termination from keyboard

Syntax

SET READEXIT ON | OFF | (<expL>)

See Also

SET EDITFIELD, SET PCKEYS, READEXIT()

Description

If READEXIT is set ON, then termination of a read, by using the [CURSOR UP] and [CURSOR DOWN] keys, is enabled. When READEXIT is set OFF, termination of a read, by using the [CURSOR UP] and [CURSOR DOWN] keys, is disabled. The command PCKEYS must be set ON for READEXIT to have any effect. By default, READEXIT is set ON. The READEXIT command is synonymous with the READEXIT() function.

Example

```
set pkeys on
set readexit on
m_var = space(10)
read
```

Products

Recital Terminal Developer

SET READINSERT

Class

Screen Forms

Purpose

Set insert mode on automatically

Syntax

SET READINSERT ON | OFF | (<expL>)

See Also

SET EDITFIELD, SET PCKEYS, EDITFIELD(), READINSERT()

Description

The READINSERT ON | OFF command allows you to toggle into insert mode automatically. If READINSERT is set ON, the form will be in insert mode when first activated. This command is a synonym for the READINSERT() function. By default, SET READINSERT is OFF and the user must toggle insert mode on with the insert key.

Example

```
set readinsert on
read
```

Products

Recital Terminal Developer

SET REFRESH

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET REFRESH TO <expN>

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

set refresh to 300

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET RELATION

Class

Table Organization

Purpose

Specify relationships between open tables

Syntax

```
SET RELATION TO [<key> INTO <alias>[,<key INTO <alias>][ADDITIVE]]  
SET RELATION ON | OFF | (<expL>)
```

See Also

CREATE VIEW, SET VIEW, SET FILTER

Description

The SET RELATION TO command defines a relationship between the currently selected table and one or more other tables. The target table of the relationship must be open prior to establishing relationships. The target table should be indexed on the <key expression> and that index should be the master index associated with the target table. If the target table is not indexed, and SET AUTOJOIN is ON, then an index will be created automatically to satisfy the relationship. Use of the SET AUTOJOIN feature should be restricted to defining temporary relationships only, as a temporary index is created each time the relationship is established. Permanent relationships can be defined with the CREATE VIEW command, and established with the SET VIEW TO command.

If the ADDITIVE keyword is specified, the specified relationship is added to the relationships already defined for the currently selected table. Twenty workareas are supported unless the environment symbol DB_MAXWKA is set higher. This allows for a maximum of DB_MAXWKA-1 relationships. Cyclic relationships are invalid and checks are automatically made for these, and an error given if one exists. The SET RELATION TO command with no relationship specified cancels any current relationships.

When a workarea is closed, or a new table is opened in a workarea, the relationships are automatically canceled. The SET RELATION ON or OFF commands enable or disable the satisfying of relationships on a system wide basis. Whenever a record is read from the active table, the <key expression> is evaluated for each of the related tables, and the resulting key is searched for in their master index files. If the key is found, the corresponding record is read from the required table into its associated workarea. Relationships from that workarea to other workareas are then satisfied.

If a key is not found in the master index file of a related table, the record buffer in the associated workarea for that particular table, and any tables related to it, are initialized in the same way as the APPEND BLANK command operates. Relationships can also be specified with @...GET commands, please see the @...GET command for details. It should be noted that the <key expression> can reference fields in any workarea by using the alias pointer notation '->' or '.'.

Example

```
select b
use addresses index add_names
select c
use accounts index acc_names
select a
use patrons index events, dates, names
set relation to name into addresses,;
    name into accounts
set filter to accounts->balance>1000
list date, event,;
    addresses->street,;
    addresses->city,;
    accounts->balance
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET REPROCESS

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET REPROCESS TO <expN> [SECONDS] | TO AUTOMATIC

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

```
set reprocess to 300
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET RETAINMENU

Class

Screen Environment

Purpose

Dictate if a menu is exited after selecting an item

Syntax

SET RETAINMENU <ON | OFF>

See Also

@...MENU, DEFINE MENU, MENU, MENU AT, MENU FORMAT, RESTORE MENU, SAVE MENU

Description

When set ON, execution of a system menu item returns to the calling menu upon completion of the action. When set OFF, the menu is exited after the action is completed. Default is off.

Example

set retainmenu on

Products

Recital Terminal Developer

SET ROLLBACK

Class

Multi-User

Purpose

Enable or disable automatic rollback

Syntax

SET ROLLBACK ON | OFF | (<expL>)

See Also

BEGIN/END TRANSACTION, ISMARKED(), RESET IN, ROLLBACK, COMPLETED(), ROLLBACK()

Description

The SET ROLLBACK ON command invokes automatic rollback and recovery in all tables affected by transactions performed within a BEGIN TRANSACTION....END TRANSACTION block. A 'rollback' causes record contents to be restored to their value before modification (e.g. at the time BEGIN TRANSACTION was issued). This is particularly useful if an error occurs during the execution of a program or if an Applications Data Dictionary validation constraint is not satisfied.

When BEGIN TRANSACTION is issued all currently open files and all files opened between BEGIN and END TRANSACTION will have Before Image Journaling invoked automatically. To disable rollback for a particular workarea, see the RESET IN command. Recital Corporation recommends the use of the SET ROLLBACK ON command to obtain automatic rollback and recovery on multi-statement transactions. By default, ROLLBACK is OFF.

Example

```
use setcomm
set rollback on
begin transaction
    delete first 15
    replace all t1 with (t2*t3)/100
    list
end transaction
if completed()
    dialog box "Transaction completed"
else
    dialog box "Errors occurred during transaction"
endif
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET RUNCLEAR

Class

Recital Terminal Developer Environment

Purpose

Enable or disable clearing of the screen after the RUN/! Commands and the RUN() function.

Syntax

SET RUNCLEAR ON | OFF | (<expL>)

See Also

SET RUNWAIT, !, RUN, RUN(), DB_RUNLOG

Description

The SET RUNCLEAR command controls whether the screen is automatically cleared, before and after execution of an Operating System command via the RUN/! Commands or the RUN() function.

With SET RUNCLEAR ON, the screen is automatically cleared, and when SET RUNCLEAR is OFF, the display is retained. This later setting allows for more efficient execution of operating system commands that do not require any screen output. By default SET RUNCLEAR is ON.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, RUNCLEAR is set ON, otherwise it is set OFF.

Example

```
set runclear off
run sysbackup > /dev/null 2>/dev/null
set runclear on
```

Products

Recital Terminal Developer

SET RUNWAIT

Class

Recital Terminal Developer Environment

Purpose

Enable or disable the 'wait message' after RUN/! Commands and RUN() function.

Syntax

SET RUNWAIT ON | OFF | (<expL>)

See Also

SET RUNCLEAR, !, RUN, RUN(), DB_RUNLOG

Description

The SET RUNWAIT command controls whether a wait message is displayed after execution of a RUN/! Command or RUN() function. By default SET RUNWAIT is OFF.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, RUNWAIT is set ON, otherwise it is set OFF.

Example

```
dialog message [Backup data tables to tape?]  
if chr(lastkey()) = [Y]  
    set runclear on  
    set runwait on  
    run tar cvf /dev/rmt0 *.db*  
    set runwait off  
    set runclear off  
endif
```

Products

Recital Terminal Developer

SET SAFETY

Class

Environment

Purpose

Determine whether existing files are overwritten without prompting for confirmation

Syntax

SET SAFETY ON | OFF | (<expL>)

See Also

PACK, REINDEX, ZAP, CREATE, SORT, TOTAL, COPY, JOIN, ERASE

Description

If SET SAFETY is ON, the user is asked whether the file should be overwritten before performing the operation. By default, SAFETY is ON

Example

use patrons

copy to orders

Orders.dbf already exists. Overwrite it? (Y/N)

Products

Recital Terminal Developer

SET SCHEDULE

Class

Screen Forms

Purpose

Enable or disable pop-up time management schedule on date fields

Syntax

SET SCHEDULE ON | OFF | (<expL>)

SET SCHEDULE TO [<.sdb filename> | (<expC>)]

See Also

@...GET, APPEND, BROWSE, CALENDAR, CHANGE, EDIT

Description

When SCHEDULE is set ON, the pop-up time management schedule is activated when the [HELP] key is pressed on a date field or memory variable. The time management schedule uses a table with the field name that the [HELP] key was pressed on. If no table of that name can be found then one is automatically created with associated index and memo file. The SET SCHEDULE TO <.sdb filename> allows you to specify the name of the schedule table that will be opened when the [HELP] key is pressed. If no file extension is specified, '.dbf' is used. The table structure of the schedule table contains 25 fields as follows:

Name	Type	Width	Description
DATE	Date	8	Current date
DETAIL8AM	Character	35	Details for 8 am
MEMO8AM	Memo	Unlimited	Full details for 8am
DETAIL9AM	Character	35	Details for 9am
MEMO9AM	Memo	Unlimited	Full details for 9am
DETAILS10AM	Character	35	Details for 10am
MEMO10AM	Memo	Unlimited	Full details for 10am
DETAILS11AM	Character	35	Details for 11am
MEMO11AM	Memo	Unlimited	Full details 11am
DETAILS12PM	Character	35	Details for 12pm
MEMO12PM	Memo	Unlimited	Full details 12pm
DETAILS1PM	Character	35	Details for 1pm
MEMO1PM	Memo	Unlimited	Full details for 1pm
DETAILS2PM	Character	35	Details for 2pm
MEMO2PM	Memo	Unlimited	Full details for 2pm
DETAILS3PM	Character	35	Details for 3pm
MEMO3PM	Memo	Unlimited	Full details for 3pm
DETAILS4PM	Character	35	Details for 4pm
MEMO4PM	Memo	Unlimited	Full details for 4pm
DETAILS5PM	Character	35	Details for 5pm
MEMO5PM	Memo	Unlimited	Full details for 5pm
DETAILS6PM	Character	35	Details for 6pm
MEMO6PM	Memo	Unlimited	Full details for 6pm
DETAILS7PM	Character	35	Details for 7pm
MEMO7PM	Memo	Unlimited	Full details for 7pm

The index key is a date that is defined in the index file of the same name of the schedule table. If EXCLUSIVE is set OFF, the time management schedule table is opened shareable with automatic record locking.

Example

set schedule to
set schedule on

Products

Recital Terminal Developer

SET SCOREBOARD

Class

Recital Terminal Developer Environment

Purpose

Enable or disable some informational messages

Syntax

SET SCOREBOARD ON | OFF | (<expL>)

See Also

SET STATUS, SET MENU

Description

If SET SCOREBOARD is ON and SET STATUS is OFF, then certain informational messages are displayed on line 1 of the terminal screen instead of appearing in the status bar. By default, SCOREBOARD is OFF.

Example

```
set status off
set scoreboard on
set menu off
use patrons index events, dates, names
append
```

Products

Recital Terminal Developer

SET SCREENCAPTURE

Class

Screen Forms

Purpose

Export Recital Terminal Developer screens for import into Internet Developer.

Syntax

SET SCREENCAPTURE ON | OFF | (<expL>)
SET SCREENCAPTURE TO <expN>

See Also

SET UICONTROLS, SET()

Description

The *File / Import Recital form...* menuitem in Internet Developer allows a Recital Terminal Developer form to be imported into Internet Developer using a special Form Definition file. These Form Definition files have an extension of '.kfm'. The Form Definition file is created from within Recital Terminal. Form Definition files can be created even if format (.fmt) files are not used for the user interface screens.

The SET SCREENCAPTURE ON | OFF command switches the capturing of user interface elements on or off. Captured definitions are saved to the Form Definition file when the F1 hotkey is pressed: a dialog box will be displayed, prompting for a filename. The SET SCREENCAPTURE TO <expN> command allows a different key (i.e. not F1) to be used as the hotkey for saving definitions. Specify the numeric INKEY value of the key to be used. To find out the INKEY value, do the following:

```
> ? inkey(0)
```

Press the key to be used

The inkey value will be returned

The kfm file will contain all the definitions of user interface elements captured since the SET SCREENCAPTURE ON command was issued. To prevent multiple screens being saved to the same file and imported into Internet Developer as a single screen, SET SCREENCAPTURE should be toggled on and off in between individual screens. The SET function can be used to check whether SCREENCAPTURE is ON (.T.) or OFF (.F.).

Definitions of the following user interface elements are saved to the kfm file:

```
@...GET  
@...SAY  
@...MENU
```

The kfm file can then be imported into Internet Developer using the *File / Import Recital form...* menuitem. A new frame is created in the existing project with the following elements:

```
@...GETs become entryfields  
@...SAYs become labels  
@...MENUs become menuitems and sub-menuitems (pull-downs) in a menu bar
```

Example

```
// Set up hotkey (tab) to toggle screenshot mode  
on key label tab set screenshot (!set([screenshot]))
```

```
// Change screencapture hotkey to F2
set screencapture to 395
// Run application, toggling screencapture and saving screens as required
do myapp
```

Products

Recital Terminal Developer

SET SCREENIO

Class

Recital Terminal Developer Environment

Purpose

Handle processing of character 255

Syntax

SET SCREENIO ON | OFF | (<expL>)

See Also

BLINK(), BOLD(), CHR(), REVERSE(), UNDERLINE(), SET CONSOLE, SET MAPCHAR, SET SCREENMAP,

Description

The SET SCREENIO command allows the programmer to handle processing of the special screen attributes character, CHR(255). Standard processing uses the screen attributes character to mark sections of the screenmap for special video attributes such as blink or reverse. The SET SCREENIO command allows for temporary disabling of the processing of CHR(255).

By default SET SCREENIO is OFF to disable special processing and display CHR (255) as a space character.

Example

```
set screenio off
? chr(255)
set screenio on
```

Products

Recital Terminal Developer

SET SCREENMAP

Class

Screen Environment

Purpose

Enable or disable automatic screen map control.

Syntax

SET SCREENMAP ON | OFF | (<expL>)

See Also

SAVE SCREEN, RESTORE SCREEN, @...SAY, @...MENU, MENU, SET TBUFSIZE, SET CONSOLE, CHR(), SET MAPCHAR, SET SCREENIO, SET DEVICE, SET PRINTER, SET ALTERNATE

Description

When SCREENMAP is set ON, an internal map is kept of the contents of the terminal screen. Refreshing only those parts of the screen that need updating optimizes all terminal output. For popup menus to be available, and SAVE/RESTORE SCREEN command to work properly, SCREENMAP should be ON. Note also that when SCREENMAP is ON, all system generated forms (as in BROWSE, APPEND, CHANGE, EDIT, QUERY, CREATE) use screen imaging.

It is recommended that SCREENMAP is left ON unless it specifically needs to be turned OFF, because screen displays are more efficient with SCREENMAP ON.

Any output to the screen from the RUN command or displays of parts of the screen with SCROLL, are not mapped. When using the SET SCROLL command, it is recommended that SCREENMAP is turned OFF.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, SCREENAMP is set ON. If a value of .F. is returned, SCREENMAP is set OFF.

NOTE: Because of the screen output optimization performed with SCREENMAP ON, output directed to the printer attached to the terminal will not print correctly. In such a case, you should turn SCREENMAP OFF before printing the report, then set SCREENMAP back ON.

Example

```
// Program to illustrate use of SET ALTERNATE commands
// '.txt.' extension will be assumed
set alternate to alt
? [Write this text to file]
set alternate off
? [Don't write this text to file]
?
set console off
set screenmap off
set device to print
set alternate on
setprc(0,0)      && set coordinates to 0,0
@1,0 say [write this to file to the file, not to the screen]
close alternate
set console on
set screenmap on
set device to screen
```

```
return  
// end of program
```

Products

Recital Terminal Developer

SET SCROLL

Class

Screen Environment

Purpose

Specify a vertical scrolling region on the screen.

Syntax

SET SCROLL TO <expN1>, <expN2> | DEFAULT

See Also

SCROLL()

Description

The SET SCROLL command allows you to specify a vertical scrolling region. The scrolling region will be displayed between rows <expN1>, the top row, and <expN2>, the bottom row. Text displayed on the screen lying outside of the scrolling region remains static on the screen.

By default the scrolling region covers the complete screen, from line 0 to line 23/24. Rows are addressable from 0 to 23/24, and columns from 0 to 79.

Example

```
set scroll to 10,19
@8, 0 say "Date Event "
@9, 0 say "-----"
@10,0
list all off date, event while inkey()=0
set scroll to default
clear
```

Products

Recital Terminal Developer

SET SECONDS

Class

Date and Time Data

Purpose

Specify whether the display of datetime values includes seconds

Syntax

SET SECONDS ON | OFF | (<expL>)

See Also

CDOW(), CMONTH(), CTOD(), DATE(), DATETIME(), DAY(), DAYS(), DMY(), DOW(), DTOC(), DTOM(), DTOS(), DTOV(), EPOCH(), GOMONTH(), MDY(), MONTH(), MTOD(), QUARTER(), STOD(), VTOD(), YEAR(), SET CENTURY, SET DATE, SET EPOCH, SET HOURS, SET MARK

Description

If SET SECONDS is ON, then the display of datetime values includes seconds. If SECONDS is OFF then the time part of datetime values only includes hours, minutes and AM | PM. By default, SECONDS is ON. The SET HOURS set command determines whether hours are shown in 24 hour format (OFF) or in 12 hour format with AM | PM postfix (ON).

Example

```
set seconds off
```

```
? datetime()
```

01/23/2004 01:18 PM

```
set seconds on
```

```
? datetime()
```

01/23/2004 01:18:22 PM

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SEPARATOR

Class

Numeric Data

Purpose

Change the character used to mark thousands

Syntax

SET SEPARATOR TO [<expC>]

See Also

SET DECIMAL, SET PRECISION, SET POINT

Description

The SET SEPARATOR TO command changes the thousand mark character from a comma to the character specified by <expC> and is used in conjunction with a picture statement or clause. This command, along with SET POINT, is typically used to change the way a decimal is written from 123,456.98 to 123.456,98. In order for the SET SEPARATOR command to work properly when outputting with @...SAYS or ?, a picture must be used that conforms to the setting of SET POINT and SET SEPARATOR. If <expC> is more than one character long, the SET SEPARATOR command will only use the first character in the string. When used without a character expression, the SET SEPARATOR command changes the thousand mark character back to the default of a comma (.).

Example

set point to “,”

set separator to “.”

@1,1 say ord_value picture “999.999.999.99”

1.234.567,89

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SEQNO

Class

Table Organization

Purpose

Reset sequence numbering to specified number

Syntax

SET SEQNO TO <expN>

See Also

ZAP, SEQNO()

Description

The SET SEQNO TO <expN> command resets the sequence number of the currently active table to the specified <expN> value. The next time the SEQNO() function is called, the value returned will be <expN> + 1. The ZAP command will reset the sequence number to 0. The next time the SEQNO() function is called, the value returned will be 1.

Example

```
use demo
set seqno to 2000
edit
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SHADOW

Class

Screen Environment

Purpose

Enable or disable shadowing of pop-up menus

Syntax

SET SHADOW ON | OFF | (<expL>)

See Also

SET BORDER, SET HIGHLIGHT, SET BLINK, SET INTENSITY

Description

When SHADOW is set ON, the screen is shaded behind the right hand side and bottom line of pop-up menus and dialog. This feature helps to distinguish the menus and boxes from the screen on which they overlay. If SHADOW is set OFF, then this feature is disabled. By default, SET SHADOW is OFF.

Example

```
set shadow on
use demo
edit
// Press the [HELP] key
```

Products

Recital Terminal Developer

SET SKIP

Class

Table Organization

Purpose

Automatically establishes one-to-many relationships across related tables

Syntax

SET SKIP TO [<alias>,<alias>]...

SET SKIP ON | OFF | (<expL>)

See Also

SET RELATION, SET FILTER

Description

The SET SKIP TO <alias> command allows for master to detail relationships between tables. It allows you to access all detail records in a related table. This command is only effective when SET RELATION is ON, and when a relationship exists between data files. The SET SKIP TO <alias> command operates by starting at the innermost relationship and scanning for all duplicate relationship keys to satisfy all detail records for the relationship. It then skips up a level in the relationship chain and performs the same operation on that data file. The record pointer in all specified related data files will update before the record pointer in the active data file is changed. Then it skips up to the master data file, reading the next record and setting up related data files.

Master to detail relationships that have been set up with the SET SKIP TO <alias> command only affect the DISPLAY, LIST and REPORT commands. The number of workareas supported is determined by the environment variable / symbol / registry entry DB_MAXWKA. By default, this is set to 20 but it can be raised, allowing up to DB_MAXWKA-1 relationships. The SET SKIP TO command disables relationships previously set with the SET SKIP <alias> command. The SET SKIP ON/OFF commands enable or disable the satisfying of the master detail relationship defined with the SET SKIP TO <alias> command on a system wide basis.

Example

```
select c
use invoices index cust_code
select b
use orders
set relation to cust_code into invoices
select a
use customers
set skip on
set relation to cust_code into invoices
set skip to order, invoices
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SOFTSEEK

Class

Indexing

Purpose

Sets soft seeking of data

Syntax

SET SOFTSEEK ON | OFF | (<expL>)

See Also

SET NEAR

Description

When SET SOFTSEEK is ON, searches of the table go to the record immediately following the potential location of the search key in the relevant file, if the key itself is not found. SET SOFTSEEK is synonymous with SET NEAR. Both SET NEAR and SET SOFTSEEK function when any valid expression is used with the SEEK command. It has no effect on relationships. By default, SOFTSEEK is OFF.

Example

```
set softseek on
use demo
select state
seek "MB"
? found()
```

.F.

```
? eof()
```

.F.

```
? state
```

MD

```
set softseek off
seek "MB"
? found()
```

.F.

```
? eof()
```

.T.

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SPACE

Class

Input/Output

Purpose

Enable or disable spaces between expression output

Syntax

SET SPACE ON | OFF | (<expL>)

See Also

?, ??

Description

The SET SPACE command affects the way that the ? and ?? commands output expressions. If SET SPACE is ON, a space is output after each expression is output. When SET SPACE is OFF, a space is not output after each expression is evaluated. The ? command outputs a carriage return, then a line feed, then evaluates each expression in turn and displays the result. The ?? command evaluates the expression and displays the results on the same line, and no carriage return/line feed sequence is output. By default, SET SPACE is ON.

Example

set space on

? "A","B",(2*3),(2+6),"C"

A B 6 8 C

set space off

? "A","B",(2*3),(2+6),"C"

AB 6 8C

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SQL

Class

Environment

Purpose

Enable or disable use of SQL statements

Syntax

SET SQL ON | OFF | (<expL>)

SET SQL TO RECITAL | VFP | MYSQL

Description

When SQL is set ON, commands that follow are assumed to be SQL, not Recital/4GL. In the development environment of Recital Terminal Developer, the interactive SQL prompt will be displayed and valid Recital/ SQL commands can be executed. Most of the Recital non-SQL commands can also be executed, but commands that conflict with SQL are not permitted. When SQL is set OFF, the normal Recital prompt is displayed. By default, SQL is OFF.

The SET SQL ON command can also be used in config.db configuration files for session, application or system wide setting. Since it affects the compilation of programs, it should be set prior to compilation rather than in a program itself. Program files with a '.sql' file extension are automatically compiled and run with SET SQL ON. When SQLDIALECT is set to Recital, embedded SQL commands within programs must be preceded by the EXEC SQL command unless SET SQL is ON.

The SET SQL TO <dialect> command is synonymous with the SET SQLDIALECT TO <dialect> command.

Example

```
> set sql on
```

```
Recital/SQL> select * from accounts;
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SQLDIALECT

Class

SQL Applications

Purpose

Configure SQL statement syntax

Syntax

SET SQL[DIALECT] TO RECITAL | VFP | MYSQL

Description

Where Recital, MySQL and VFP differ in their SQL syntax, the SET SQLDIALECT command can be used to select the syntax to be used.

By default, SET SQLDIALECT is set to RECITAL. Since the SET SQLDIALECT setting affects program compilation, it should be set prior to compilation rather than in a program itself, for example in a config.db configuration file.

NOTES:

;

The semi-colon, ';', is used to terminate SQL statements when SQLDIALECT is set to RECITAL or MYSQL. It is used as a line continuation character when SQLDIALECT is set to VFP.

EXEC SQL

When SQLDIALECT is set to RECITAL, SQL statements embedded in programs must be preceded by the EXEC SQL statement unless SET SQL is ON.

.sql programs

When a program with a '.sql' file extension is run, SQLDIALECT is automatically set to MYSQL and SET SQL is set ON.

Example

```
set sqldialect to vfp
```

```
set sql to mysql
```

```
set sql on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SQLPROMPT

Class

Recital Terminal Developer Environment

Purpose

Specify the Recital/SQL interactive prompt

Syntax

SET SQLPROMPT TO <expC> | DEFAULT

See Also

SET SQL

Description

The SET SQLPROMPT TO <expC> command allows you to specify the interactive SQL prompt. The <expC> can be a maximum of 10 characters long. The SET SQLPROMPT TO DEFAULT command sets the SQL prompt to 'SQL>'.

Example

```
Recital/SQL> set sqlprompt to "R-SQL> "  
R-SQL> set sqlprompt to default  
SQL>
```

Products

Recital Terminal Developer

SET SQLROWID

Class

SQL Applications

Purpose

To include a unique row identifier in SELECT * statements

Syntax

SET SQLROWID ON | OFF

See Also

SELECT, UNIQUEROWID()

Description

The SET SQLROWID ON | OFF command is used to determine whether a unique row identifier in should be included in SELECT * statements. If SET SQLROWID is ON, the unique row identifier will be included, if SET SQLROWID is OFF, only the fields from the table will be included.

Example

```
set sql on
set sqlrowid on
select * from state.rdb where state = "M";
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET STATUS

Class

Recital Terminal Developer Environment

Purpose

Determines whether the status bar is displayed on line 23 of the screen

Syntax

SET STATUS ON | OFF | (<expL>)

See Also

SET MESSAGE

Description

If SET STATUS is ON, then the system displays a status bar on line 23 of the screen. The status bar contains information regarding the current workarea. The status bar is divided into six sections. Whenever possible, you should SET STATUS ON, as the status bar contains a great deal of useful information. If STATUS is OFF, messages output with the SET MESSAGE command are not displayed in the message line.

Example

```
set status off
use patrons index events, dates, names
set format to bigform
change
set status on
set format to normalform
change
```

Products

Recital Terminal Developer

SET STEP

Class

Error Handling and debugging

Purpose

Single step through the execution of a program

Syntax

SET STEP ON | OFF | (<expL>)

See Also

CANCEL, DEBUG, DISPLAY MEMORY, DISPLAY STATUS, DO, LIST MEMORY, LIST STATUS, QUIT, RESUME, SUSPEND, SET COMPILE, SET DEBUG, SET DEVELOPMENT, SET ECHO, SET TALK

Description

This command provides a pop-up debugger for debugging programs. The commands COMPILE and DEVELOPMENT must be set OFF when using SET STEP ON. The SET STEP ON command displays a pop-up debugger with the specified program or procedure. The pop-up program debugger can also be activated with the DEBUG command. When the DEBUG command is used, SET COMPILE does not have to be OFF as this is handled automatically. You must, however, load procedure libraries with SET COMPILE OFF when you are debugging a procedure. By default, STEP is OFF.

Example

```
set echo on
set step on
do testprg
```

Products

Recital Terminal Developer

SET STRINGBUF

Class

String Data

Purpose

Specify or reset the memory used for temporary string operations.

Syntax

SET STRINBUF TO<expN>

See Also

AT(), ATNEXT(), SUBSTR(), STR(), STUFF(), STRZERO(), MTOS(), RAT()

Description

This command allows you to specify or reset the amount of memory needed for temporary string operations used in character expressions. The default string buffer size is 8192 bytes.

Example

```
// Reset string buffer  
set stringbuf to 8192
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET STRINGLEN

Class

Expressions and Type Conversion

Purpose

Affects the size of the return value of the STR() function

Syntax

SET STRINGLEN ON | OFF | (<expL>)

See Also

STR()

Description

This set command affects the behavior of the STR() function. If SET STRINGLEN is ON and the optional length parameter of the STR() function is not specified, STR() will return a string the same length as the numeric column. This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, STRINGLEN is set ON. If a value of .F. is returned STRINGLEN is set OFF. By default STRINGLEN is OFF.

NOTE: this command has no effect on the STRZERO() function.

Example

```
? len(total)
```

5

```
? len(str(total))
```

10

```
set stringlen on
```

```
? len(str(total))
```

5

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SYSLOGGING

Class

Environment

Purpose

Enable or disable internal system logging

Syntax

SET SYSLOGGING ON | OFF | (<expL>)

See Also

ERROR(), MESSAGE(), DB_LOGDIR

Description

When SET SYSLOGGING is ON internal system logging is performed while the process is running. The information logged can be used to find performance problems or track down system errors. The format of the log file is as follows:

Column Number	Description
One	Process ID of the user writing the line to the log file
Two	Date line was written
Three	Time line was written
Four	Internal file name
Five	Internal line number
Six	Operating system error number
Seven	General description

All the logging information is written to a file called recital.log. If the environment variable *DB_LOGDIR* is not defined, then the log files are created in the root directory, otherwise the log files will be created in the specified directory.

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, SYSLOGGING is set ON. If a value of .F. is returned, SYSLOGGING is set OFF. By default SYSLOGGING is OFF.

Example

```
set syslogging on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET SYSMENU

Class

Recital Terminal Developer Environment

Purpose

Enable or disable the system menu

Syntax

SET SYSMENU ON | OFF | (<expL>) |AUTOMATIC

See Also

SET COMMANDWINDOW

Description

The SET SYSMENU command is used to enable or disable the system menu bar which displays along the top of the screen when you are at the interactive prompt. When SYSMENU is set OFF from the interactive prompt, the menu bar cannot be accessed with the [MENUBAR] key. When SYSMENU is set OFF from a config.db file, the menu bar is not displayed. When SYSMENU is set ON, pressing the [MENUBAR] key accesses the menu bar. If the menu bar has been disabled in a config.db file, SET SYSMENU ON will display the menu bar and enable the [MENUBAR] key to access it.

AUTOMATIC

The AUTOMATIC setting ensures that the system menu bar is visible during program execution, allowing access to the appropriate menu items according to the current command.

Example

```
set sysmenu off
```

Products

Recital Terminal Developer

SET TALK

Class

Recital Terminal Developer Environment

Purpose

Determine whether command results will be echoed to the screen

Syntax

SET TALK ON | OFF | (<expL>)

See Also

SET ODOMETER, SET ECHO, SET STEP, SET DEBUG

Description

The SET TALK ON command can be used to assist in program debugging. If SET TALK is ON, then the result of certain commands will be echoed to the screen. Certain commands which may take a long time, such as PACK, INDEX, REINDEX etc. count on the screen to inform you of their progress. If TALK is OFF, this counting is disabled. The SET ODOMETER command can be used in conjunction with the SET TALK command to adjust the rate at which the counter is updated. By default, TALK is ON.

Example

```
set talk on
i = 10
```

10

```
set talk off
```

```
i = 10
```

```
set talk on
```

```
set odometer to 10
```

```
use patrons
```

```
index on event to events
```

Indexing complete, 2400 records indexed.

Products

Recital Terminal Developer

SET TBUFSIZE

Class

Recital Terminal Developer Environment

Purpose

Specify size of terminal output buffer to optimize terminal I/O throughput.

Syntax

SET TBUFSIZE TO <expN>

See Also

SET ICACHE, SET DCACHE, SET PCACHE, SET SCREENMAP

Description

The SET TBUFSIZE TO <expN> command allows for the specification of the size of the terminal output buffer. All output to the terminal is stored in this output buffer until either the buffer is full, or a terminal read is requested. Certain commands can take a long time to automatically flush the terminal output buffer.

A large value for this setting enables forms to be displayed using one Operating System write request, thereby minimizing Operating System overhead, and taking advantage of any DMA communications controllers. The form is also displayed on the screen more smoothly.

On OpenVMS, the maximum setting of TBUFSIZE depends on the value of the SYSGEN parameter MAXBUF and the maximum buffered I/O quota in the user authorization entry for the user. OpenVMS Developer inspects the size of the MAXBUF parameter and sets TBUFSIZE as high as possible. The OpenVMS Systems Manager can make adjustments so that OpenVMS Developer users can take advantage of an increased TBUFSIZE. To take full advantage of a large TBUFSIZE on OpenVMS, the terminal should be set to DMA, using the DCL command:

```
$ SET TERMINAL/DMA
```

Example

```
set tbufsize to 3000
```

Products

Recital Terminal Developer

SET TCACHE

Class

SQL Applications

Purpose

Enable, disable and configure table caching during SQL operations

Syntax

SET TCACHE ON | OFF | TO <expN>

See Also

ALTER INDEX, ALTER TABLE, CREATE INDEX, CREATE TRIGGER, DELETE, DROP INDEX, GRANT, INSERT, REVOKE, SELECT, UPDATE

Description

The SET TCACHE command is used to enable or disable table caching and configure the number of tables that can be cached during SQL operations. If SET TCACHE is ON, a table accessed by an SQL statement is left open until the session or connection is closed. With SET TCACHE OFF, tables are opened and closed with every SQL statement.

With SET TCACHE ON, the default number of tables that can be cached corresponds to the number of available workareas (DB_MAXWKA environment variable / symbol). This number can be reduced using the SET TCACHE TO <expN> command. The <expN> specifies the maximum number of tables that can be cached.

Enabling TCACHE can give significant performance benefits where multiple operations are being carried out on the same table or tables. With TCACHE ON, individual tables can still be closed if required using the USE command in the relevant workarea or the CLOSE <alias> command.

Example

// Up to 4 tables will remain open after a completed SQL statement

```
set tcache on
set tcache to 4
set sql to recital
```

EXEC SQL

```
OPEN DATABASE southwind;
```

EXEC SQL

```
INSERT INTO shippers
VALUES (4, "Recital Corporation", "(978) 921-5594");
```

EXEC SQL

```
UPDATE employees
SET extension="256"
WHERE employeeid=4;
```

EXEC SQL

```
DELETE FROM suppliers
WHERE supplierid=1;
```

```
EXEC SQL
  ALTER TABLE example
  ADD (email char(40));
```

```
EXEC SQL
SELECT * from products;
```

```
// Next free workarea is workarea 5
// as tables in workareas 1 to 4 remain open
? workarea()
?
close databases
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TEDIT

Class

Recital Terminal Developer Environment

Purpose

Specify an alternate text editor to be used when MODIFY COMMAND is issued.

Syntax

SET TEDIT TO <expC>

See Also

MODIFY COMMAND, ALIAS, SPAWN, RUN, SET WP

Description

With the SET TEDIT TO <expC> command, you can specify an alternate text editor which will be used when you issue the MODIFY COMMAND <filename> command.

Example

set tedit to "wmc"

set tedit to "edit/nojournal"

Products

Recital Terminal Developer

SET TERMINAL

Class

Recital Terminal Developer Environment

Purpose

Specify the terminal being used.

Syntax

SET TERMINAL TO <terminal-name>

See Also

GETENV(), PUTENV()

Description

The SET TERMINAL TO <terminal-name> command specifies the type of terminal that is being used. This command is normally only used for testing purposes. When Recital Terminal Developer starts, it will automatically issue a SET TERMINAL TO command depending upon the contents of the environment variable/symbol, DB_TERM. This environment variable (UNIX/Linux), or symbol (OpenVMS) is defined in the login file (profile.db / login.com).

Example

set terminal to ansi

Products

Recital Terminal Developer

SET TEXTMERGE

Class

Input/Output

Purpose

Specify delimiting characters

Syntax

```
SET TEXTMERGE ON | OFF | (<expL>)  
SET TEXTMERGE TO [<filename>] [ADDITIVE]  
SET TEXTMERGE DELIMITERS TO [<expC1>] [,<expC2>]
```

See Also

\ and \[, TEXT...ENDTEXT

Description

The SET TEXTMERGE ON | OFF command determines whether expressions that have been enclosed by the text merge delimiters will be evaluated, or output literally. When SET TEXTMERGE is ON, delimited expressions are evaluated and then output to the screen or to a file. If SET TEXTMERGE is OFF, delimited expressions are output literally. Delimited expressions may include table field names, memory variables, and functions. The default text delimiters are << and >>. To use a different set of characters to delimit text, use the SET TEXTMERGE DELIMITERS command. Delimited text is output when it is placed after the \ and \[commands, or when it is between the TEXT...ENDTEXT commands. When SET TEXTMERGE is ON, expressions placed within the TEXT...ENDTEXT commands are evaluated and merged with text.

The SET TEXTMERGE TO command directs delimited text to an output file. If the name of the text file, as specified by <filename>, already exists, the file will be overwritten unless the ADDITIVE keyword is used. If the text file does not exist and the ADDITIVE qualifier is used, the file will be created. If the file does exist, then the text will be appended to the end of the text file. If the command SET TEXTMERGE TO is issued without a <filename>, all output will be directed to the current output device, such as the screen or a window.

The SET TEXTMERGE DELIMITERS TO command is used to specify the delimiting characters. The first character expression, <expC1>, is used to specify the beginning text delimiter. If only one character expression is specified, the same character is used to delimit the beginning and ending of text. The second character expression, <expC2>, is used to specify the ending text delimiter. If no character expressions are specified, the SET TEXTMERGE DELIMITER TO command restores the delimiting characters to << and >>.

Example

```
use test  
set textmerge to balance.txt  
et textmerge to delimiters to “{“,”}”  
set textmerge on  
go top  
do while balance > 0  
\Date: {{date()}}  
\Name: {{last_name}} {{first_name}}  
\  
\Account Number : {{account_no}}  
\Current Balance : {{balance}}
```



```
\Credit Limit:      : {{limit}}  
\
```

```
enddo  
set textmerge off  
set textmerge to
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TIME

Class

Recital Terminal Developer Environment

Purpose

Determine whether the current time is displayed whenever in the interactive command prompt

Syntax

SET TIME ON | OFF | (<expL>)

See Also

SET PROMPT

Description

If SET TIME is ON, then the current system time is displayed in the format HH:MM:SS before each interactive command prompt. By default, TIME is OFF.

Example

```
set time on
16:28:04 > use patron index events, dates, names
16:29:08 > reindex
16:41:04 >
```

Products

Recital Terminal Developer

SET TIMEOUT

Class

Screen Environment

Purpose

Set the timeout delay

Syntax

SET TIMEOUT [TO <expN> | <ON | OFF>]

See Also

ON TIMEOUT, SET CLOCK, SET CLOCKDISPLAY

Description

The SET TIMEOUT command sets the number of seconds (<expN>) before the event assigned by the ON TIMEOUT command is executed. SET TIMEOUT ON activates the timer, SET TIMEOUT OFF deactivates the timer. SET CLOCK must be ON.

Example

```
procedure p_timeout
    dialog box "Timeout Occurred."
    set timeout off
    quit
return

set clock on
on timeout do p_timeout
set timeout to 4
set timeout on
m_var = space(10)
@00,00 say "Enter Value: " get m_var
read
return
```

Products

Recital Terminal Developer

SET TITLE

Class

Information Center

Purpose

Enable or disable the File Description dialog box in the Recital Information Center.

Syntax

SET TITLE ON | OFF | (<expL>)

See Also

SET CATALOG, SET DESIGN, DESIGN, INFO

Description

The SET TITLE command controls the display of the File Description dialog box in the Recital Information Center. When SET TITLE is ON, and a catalog is open, a dialog box entitled 'SPECIFY FILE DESCRIPTION' appears after a new file is created. The dialog box contains a field labeled 'Title:' in which to enter a descriptive line of text of up to 80 characters. When SET TITLE is OFF, the File Description dialog box is not displayed.

File descriptions for files in a catalog may be added or modified by using the *Edit file descriptions...* menu option.

Example

```
set title on  
design
```

Products

Recital Terminal Developer

SET TMPDIR

Class

Environment

Purpose

To specify the path to be used by the SYS(3) temporary file name function

Syntax

SET TMPDIR TO <expC>

See Also

DB_TMPDIR, SYS(3)

Description

The SET TMPDIR command is used to specify the path to be used by the SYS(3) temporary file name function. The SET TMPDIR command can temporarily override the DB_TMPDIR setting that is set at the beginning of a session and cannot be modified during the session. If neither DB_TMPDIR nor SET TMPDIR are set, the SYS(3) function will return a file name without any path information. With DB_TMPDIR or SET TMPDIR set to a valid path, the return value of SYS(3) will include this path information. Please note that the <expC> should include a trailing directory separator.

Example

```
set tmpdir to /usr/tmp/  
? sys(3)  
/usr/tmp/000cf30006
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TRACEWINDOW

Class

Error Handling and Debugging

Purpose

Enable or disable the use of trace windows

Syntax

SET TRACEWINDOW ON | OFF | (<expL>)

See Also

ACTIVATE WINDOW, DEFINE WINDOW, SET ERRORWINDOW, SET COMMANDWINDOW

Description

The SET TRACEWINDOW command is used to enable or disable the use of trace windows. A trace window is a window that automatically activates itself and displays each executing line of a currently running program. Trace windows are specified by including the TRACE keyword with the DEFINE WINDOW command. A window is an area of the screen designated for output and input. There is no limit to the number of windows you may define. Windows are defined with the DEFINE WINDOW command, and are activated with the ACTIVATE WINDOW command.

To display lines of the executing program, a trace window requires the following command settings: SET COMPILE OFF, SET HISTORY ON, SET DOHISTORY ON. When SET TRACEWINDOW is ON, trace windows are automatically activated when a program is run. When SET TRACEWINDOW is OFF, trace windows do not activate or display when programs are run.

Example

```
// Entered interactively
set compile off
set history on
set dohistory on

// In program file
define window debug;
    from 15,01 to 22,78;
    trace
set tracewindow on

do progname
```

Products

Recital Terminal Developer

SET TRANSACTION

Class

SQL Applications

Purpose

To set the current transaction state

Syntax

SET TRANSACTION [READ ONLY | READ WRITE]

SET TRANSACTION ISOLATION LEVEL <level>

See Also

CLOSE, DECLARE CURSOR, DROP CURSOR, FETCH, OPEN, SELECT

Description

A current transaction state can be either read-only or read-write. Three further aspects of transaction behavior are configurable: dirty reads, non-repeatable reads and phantom reads. Dirty reads occur when a transaction updates a row, then a second transaction reads that row before the first transaction commits. If the first transaction rolls back the change, the information read by the second transaction becomes invalid. Non-repeatable reads occur when a transaction reads a row then another transaction updates the same row. If the second transaction commits, subsequent reads by the first transaction get different values than the original read. Phantoms occur when a transaction reads a set of rows that satisfy a search condition and then another transaction updates, inserts, or deletes one or more rows that satisfy the first transaction's search condition. In this case, if the first transaction performs subsequent reads with the same search condition, it reads a different set of rows.

The <level> can be any one of the following:

- SERIALIZABLE
- REPEATABLE READ
- READ COMMITTED
- READ UNCOMMITTED

If you use a SET TRANSACTION statement, it must be the first statement in your transaction.

NOTE: This command can also be used as a standard SET COMMAND in the config.db file, to set the transaction state on a system or application wide basis.

Keywords	Description
READ ONLY	Set the default transaction type to read-only.
READ WRITE	Set the default transaction type to read-write.
ISOLATION LEVEL	Specify how the transaction will perform.
SERIALIZABLE	This will disable dirty reads, non-repeatable reads and phantom reads. This is the default isolation level.
REPEATABLE READ	This will disable dirty reads, non-repeatable reads and enable only phantom reads.
READ UNCOMMITTED	This will enable dirty reads, non-repeatable reads and phantom reads.
READ COMMITTED	This will disable dirty reads and enable non-repeatable reads and phantom reads.

Example

set transaction isolation level read uncommitted

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TRAP

Class

Xbase Compatibility

Purpose

Language compatibility only

Syntax

SET TRAP ON | OFF | (<expL>)

See Also

SET COMPATIBLE, SET FILETYPE, SET INDEXEXT, SET MEMOEXT, SET CLIPPER

Description

This command has been added for language compatibility only.

Example

```
set trap on
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TSLENGTH

Class

Text Searching

Purpose

Specify size of Text Search Index key

Syntax

SET TSLENGTH TO <expN>

See Also

INDEX, KEY(), LOWER(), NDX(), TSPOS(), TSWORD(), UPPER(), SET INDEX, SET ORDER, DB_TSINDEX

Description

The SET TSLENGTH set command determines the size of Text Search Index keys. The default is 32 characters. The value to set is specified in <expN>. Smaller keys may result in inaccurate searches if the data is truncated.

The actual stored key value will be 10 bytes larger than the TSLENGTH to allow for additional text search information in the key.

The UPPER() or LOWER() functions can be used when creating indexes and when conducting index searches to make them case-insensitive.

The DB_TSINDEX environment variable / symbol must be set to “ON” when building or using Text Search Indexes.

Example

```
set tslength to 15
use example
index on tsword(first_name,1) to shortname
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET TYPEAHEAD

Class

Keyboard Events

Purpose

Specify the number of characters that the terminal typeahead buffer can contain

Syntax

SET TYPEAHEAD TO <expN>

See Also

ON KEY, INKEY()

Description

The SET TYPEAHEAD TO <expN> commands allows you to specify the maximum number of characters that can be stored in the terminal typeahead buffer. Setting the typeahead buffer to 0 disables the typeahead buffer. When typeahead is 0, the INKEY() function and the ON KEY command are also disabled.

Example

set typeahead to 10

set typeahead to 0

Products

Recital Mirage Server, Recital Terminal Developer

SET UDFPARMS

Class

Environment

Purpose

Specify whether parameters are passed to a user-defined function by value or by reference

Syntax

SET UDFPARMS TO VALUE | REFERENCE

See Also

DO, FUNCTION, PARAMETER, SET PROCEDURE

Description

By default, parameters are passed to UDFs by value. By setting UDFPARMS to REFERENCE, parameters will be passed to the UDF by reference. Variables that are passed by reference can have their values changed in the UDF, and the new values will be available at the scope where the variables were defined.

Example

```
set udfparms to value
count = 10
// Call a UDF that changes the value of count to 30
changecount(count)
? count
    10
set udfparms to reference
count = 10
// Call a UDF that changes the value of count to 30
changecount(count)
? count
    30
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET UICONTROLS

Class

Input/Output

Purpose

Determine whether user interface commands are ignored

Syntax

SET UICONTROLS ON | OFF | (<expL>)

See Also

SET SCREENCAPTURE, SET()

Description

This command determines whether commands applicable only in user interface based Recital products (Recital Terminal Developer and Recital Mirage client) are ignored. It applies to commands such as @...GET and @...SAY. Stored Procedures (server side programs [.prg]) run by the Recital Database Server will run, by default, with SET UICONTROLS OFF. This means that commands that relate to the user interface do not cause errors, but are simply ignored. By default SET UICONTROLS is ON in Recital Terminal Developer and Recital Mirage Server.

Example

```
// Running under Recital Database Server
// @...SAY will be ignored
// Recital Mirage Client or Recital Terminal Developer
// Users will see the text
use demo
@1,1 say "Reindexing tables...."
reindex all
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET UNDERLINE

Class

Screen Forms

Purpose

Determine whether fields displayed in forms should be underlined

Syntax

SET UNDERLINE ON | OFF | (<expL>)

See Also

SET INTENSITY, @...GET, READ

Description

If SET UNDERLINE is ON, then fields in the full screen forms used in commands such as APPEND, CHANGE, EDIT, INSERT, QUERY and READ will be underlined on the screen. You would normally choose to either underline the fields, or highlight them in reverse video using the SET INTENSITY command. By default, UNDERLINE is OFF.

Example

```
set intensity off
set underline on
use patrons index events, dates, names
append
```

Products

Recital Terminal Developer

SET UNIQUE

Class

Indexing

Purpose

Determine whether duplicate keys are to be included in index files

Syntax

SET UNIQUE ON | OFF | (<expL>)

See Also

INDEX, REINDEX, UNIQUE

Description

If SET UNIQUE is ON when an index file is created or recreated, records with duplicate keys are discarded. The records still exist in the table, but there is no entry for them in the index file, so they are 'hidden' from view. The unique status of the keys is stored in the index file, so whenever the index is associated with the table subsequently, the uniqueness of keys that are added to the index file is enforced.

If an attempt is made to add a duplicate key to a unique index file, the operation will not be able to complete. If a record is being appended or changed in the table through a full screen forms based command such as APPEND or CHANGE, then the system displays an appropriate message. The form cannot be exited until data entry is abandoned by pressing the [ABANDON] key, or a unique key is entered. If an attempt is made to update a record through the REPLACE command, an error message will be generated. This can be trapped with the ON ERROR command.

The UNIQUE keyword can also be included in the INDEX or REINDEX commands to provide the same effect as SET UNIQUE ON. By default, UNIQUE is OFF.

Example

```
use patrons
set unique on
index on name to persons
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET UPDATE

Class

Multi-User

Purpose

Specify forms mode to be query or update

Syntax

SET UPDATE ON | OFF | (<expL>)

See Also

BROWSE, CHANGE, QUERY, SET QUERYMODE, GETGID(), GETUID(), ACCESS()

Description

If SET UPDATE is ON, current records are locked for updating in CHANGE, EDIT or BROWSE screen editing forms. This prevents other users from updating current records at the same time. When in a form, records can be locked for update by pressing the [UPDATE MODE] key in the form. While scanning through the records in a table, particularly in the BROWSE command, it is advantageous to toggle between update mode and query mode. When update mode is ON, 'Upd' is displayed in the third section of the status bar. If the currently selected table, is open with read only access, then 'RO' is displayed in the third section of the status bar. Attempts to toggle into update mode will cause the terminal bell to sound.

You can use the SET UPDATE OFF command in conjunction with the SET QUERYMODE command to provide read only access to a table. The GETGID() function can also be used with these commands to identify the group to which a user belongs before restricting access to a table. By default, UPDATE is ON.

Example

```
if getgid()>10
    set update off
    set querymode off
endif
browse
```

Products

Recital Terminal Developer

SET VALIDATE

Class

Screen Forms

Purpose

Signify that data entered in a field on a form has passed validation checks

Syntax

SET VALIDATE ON | OFF | (<expL>)

See Also

@...GET, SET FIELDVAL, MENU AT

Description

The SET VALIDATE ON command is used to signify that data entered in a field on a form, which was specified with the @...GET VALIDATE WITH <procedure> command, has passed validation checks. The SET FIELDVAL command can be used to modify the contents of the field on the form. The validation procedure will only be called if data is entered in the field on the form, or the [HELP] key is pressed. See @...GET VALIDATE WITH for full details.

Example

```
procedure check_event
parameter s
select b
seek s
if not found()
    set fieldval to "BALLET"
endif
select a
set validate on
return
```

```
@10,10 say "Event ";
    get event;
    must_enter;
    validate with check_event
read
```

Products

Recital Mirage Server, Recital Terminal Developer

SET VAXTIME

Class

Date and Time Data

Purpose

Toggle display of VAX date time string

Syntax

SET VAXTIME ON | OFF | (<expL>)

See Also

CREATE BRIDGE, DTOV(), VTOD()

Description

The SET VAXTIME command enables or disables display of the time string portion of a VAX date. VAX dates are most commonly used with the RMS Replaceable Database Driver (RMS Bridge).

This command allows the optional logical expression <expL> to be evaluated. If a value of .T. is returned, VAXTIME is set ON. If a value of .F. is returned, VAXTIME is set OFF. Note that <expL> should be enclosed in round brackets. By default, VAXTIME is OFF.

Example

```
dDate = { 10/10/2002 }
```

```
set vaxtime on
```

```
? dtov(dDate)
```

```
10-OCT-2002 00:00:00.00
```

```
set vaxtime off
```

```
? dtov(dDate)
```

```
10-OCT-2002
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer (All OpenVMS only)

SET VERIFY

Class

Screen Environment

Purpose

Enable or disable dialog box display

Syntax

SET VERIFY ON | OFF | (<expL>)

See Also

DIALOG MESSAGE, SET DIALOG, SET HELP, SET MESSAGE

Description

The SET VERIFY command is used to enable or disable dialog box display upon execution of [ABANDON], [DELETE RECORD], [EXIT/SAVE], or [CARRY MODE] operations. If VERIFY is set ON, when a user presses a key to initiate an [ABANDON], [DELETE RECORD], [EXIT/SAVE] or [CARRY MODE] operation, a dialog box displays asking for confirmation of the operation. The dialog box contains a YES button and a NO button. The user may move the cursor to a YES or NO button and press [RETURN] to reply. Pressing the [ABANDON] key or selecting the NO button returns the user to the previous screen. Selecting the YES button continues the operation. By default, SET VERIFY is ON.

Example

```
set verify on
use test
```

Products

Recital Terminal Developer

SET VIEW

Class

Table Organization

Purpose

Establish a view previously defined with the CREATE VIEW command

Syntax

```
SET VIEW TO <.vue filename> | (<expC>)
```

See Also

CREATE VIEW, SET RELATION, ALIAS

Description

The SET VIEW TO <.vue filename> command establishes a logical view of a group of tables, which can then be viewed as one if they are related together. The filename can be substituted with a <expC>, enclosed in round brackets, which returns a valid filename. If no file extension is specified, '.vue' is used. Once a view has been established, the workarea that was current when the view was created is selected.

Example

```
use patrons index events, dates, names
// View patrons.vue defines the above command
set view to patrons
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer

SET WINDOW OF BROWSE

Class

Screen Windows

Purpose

Designate a window to display the BROWSE work surface

Syntax

SET WINDOW OF BROWSE [IN <alias>] TO <window-name>

See Also

ACTIVATE WINDOW, ACTIVATE SCREEN, CLEAR WINDOW, DEACTIVATE WINDOW, DEFINE WINDOW, HIDE WINDOW, MOVE WINDOW, MODIFY MEMO, RELEASE WINDOW, RESIZE WINDOW, RESTORE WINDOW, SAVE WINDOW, SHOW WINDOW, SET COMMANDWINDOW, SET ERRORWINDOW, SET TRACEWINDOW, SET WINDOW OF MEMO, SET WINDOW OF RELATION, SET WINDOW OF EDIT, WEXIST(), WINDOW(), WVISIBLE(), WONTOP(), WOUTPUT()

Description

The SET WINDOW OF BROWSE command designates the window used to display the default BROWSE work surface when BROWSE is invoked. The window must be defined with the DEFINE WINDOW command, however, the window does not need to be activated, as the SET WINDOW OF BROWSE command will activate the window.

The optional IN <alias> qualifier may be used to specify the workarea that SET WINDOW OF BROWSE is to work in. If this qualifier is not used, the appropriate workarea must be selected before the SET WINDOW OF BROWSE command is issued.

The BROWSE work surface will be sized and positioned to the coordinates described with the DEFINE WINDOW command. If you wish the BROWSE work surface to appear inside the window, with the border intact, use the ACTIVATE WINDOW command.

Example

```
define window browse
  from 1,1 to 11,42;
  title "BROWSE WINDOW"
set window of browse to browse
use demo
browse
```

Products

Recital Terminal Developer

SET WINDOW OF EDIT

Class

Screen Windows

Purpose

Specify the name of the window to display split screen record editing

Syntax

SET WINDOW OF EDIT [IN <alias>] TO <window-name>

See Also

ACTIVATE SCREEN, ACTIVATE WINDOW, CLEAR WINDOWS, DEACTIVATE WINDOW, DEFINE WINDOW, HIDE WINDOW, MOVE WINDOW, MODIFY MEMO, RELEASE WINDOWS, RESIZE WINDOW, RESTORE WINDOW, SAVE WINDOW, SHOW WINDOW, SET ERRORWINDOW, SET COMMANDWINDOW, SET TRACEWINDOW, SET STATUS, SET WINDOW OF MEMO, WROWS(), WCOLS(), WEXIST(), WVISIBLE(), WONTOP(), WOUTPUT()

Description

The SET WINDOW OF EDIT command designates a pre-defined window to display a default editing work surface when BROWSE is invoked in a window. A window is an area of the screen designated for output and input. Windows are defined with the DEFINE WINDOW command, and are activated with the ACTIVATE WINDOW command. There is no limit to the number of windows you may define. After the SET WINDOW OF EDIT command is issued and BROWSE is invoked, a default edit work surface will be displayed in <window-name>. When the BROWSE window is active, control can be passed to the EDIT window by pressing the BROWSE [WINDOW] key. The <window-name> is the name of the window as specified with the DEFINE WINDOW command.

The optional qualifier IN <alias> may be used to designate an edit window in another workarea. If the IN <alias> qualifier is not use, the correct workarea must be selected before issuing the SET WINDOW OF EDIT command.

Example

```
define window wEdit;  
    from 1,44 to 21,78;  
    title "EDIT WINDOW"  
use demo  
set window of edit to wEdit  
edit
```

Products

Recital Terminal Developer

SET WINDOW OF MEMO

Class

Screen Windows

Purpose

Specify the name of the window in which to edit memo fields

Syntax

SET WINDOW OF MEMO [IN <alias>] TO <window-name>

See Also

ACTIVATE SCREEN, ACTIVATE WINDOW, CLEAR WINDOWS, DEACTIVATE WINDOW, DEFINE WINDOW, HIDE WINDOW, MOVE WINDOW, MODIFY MEMO, RELEASE WINDOWS, RESIZE WINDOW, RESTORE WINDOW, SAVE WINDOW, SHOW WINDOW, SET ERRORWINDOW, SET COMMANDWINDOW, SET TRACEWINDOW, SET STATUS, SET WINDOW OF BROWSE, SET WINDOW OF EDIT, SET WINDOW OF RELATION, WROWS(), WCOLS(), WEXIST(), WINDOW(), WVISIBLE(), WONTOP(), WOUTPUT()

Description

The SET WINDOW OF MEMO TO command designates a pre-defined window in which to edit the memo of the current record. A window is an area of the screen designated for output and input. Windows are defined with the DEFINE WINDOW command, and are activated with the ACTIVATE WINDOW command. There is no limit to the number of windows you may define. The <window-name> is the name of the window as specified with the DEFINE WINDOW command. The SET WINDOW OF MEMO TO <window-name> command must be used before the MODIFY MEMO command is issued. The MODIFY MEMO command is used to edit memos in a window rather than in a pop-up notepad. After the SET WINDOW OF MEMO command is issued and BROWSE is invoked, the memo for the current record will be displayed in <window-name>. When the BROWSE window is active, control can be passed to the memo window by placing the cursor on the memo field in BROWSE and then pressing the [HELP] or [INSERT] key.

The optional IN <alias> qualifier may be used to designate a memo window in another workarea. If the IN <alias> qualifier is not used, the correct workarea must be selected before issuing the SET WINDOW OF MEMO command.

Example

```
define window wMemo
    from 13,1 to 21,42;
    title "MEMO WINDOW"
define window wBrowse;
    from 1,1 to 11,42;
    title "BROWSE WINDOW"

use demo
set window of memo to wMemo
set window of browse to wBrowse
browse
```

Products

Recital Terminal Developer

SET WINDOW OF RELATION

Class

Screen Windows

Purpose

Activate a relational window

Syntax

```
SET WINDOW OF RELATION [IN <alias>] TO [<window-name>]  
[COMMAND EDIT | BROWSE [<qualifier>]  
[DO <procedure | program>]
```

See Also

SET WINDOW OF BROWSE, SET WINDOW OF EDIT, SET WINDOW OF MEMO, DEFINE WINDOW, EDIT, BROWSE

Description

The SET WINDOW OF RELATION command activates a window to view and modify data that is related to an active BROWSE or EDIT window.

As the record pointer is moved through the BROWSE or EDIT window, data in <window> is refreshed to show only those records that are related to the parent table. The <window> must be defined with a DEFINE WINDOW command. The cursor may be moved into the related window by using the [FIND NEXT] key to cycle through all related windows. When the window is selected, any operation that is allowed with the type of work surface may be used.

The optional IN <alias> qualifier may be used to specify the workarea that SET WINDOW OF RELATION is to work in. If this qualifier is not used, the appropriate workarea must be selected before the SET WINDOW OF RELATION command is issued.

Each workarea may have a SET WINDOW OF RELATION defined. It is required that the SET WINDOW OF RELATION command be used to define the relations between the workareas before the SET WINDOW OF RELATION command is issued.

The COMMAND qualifier specifies whether data will be presented in a BROWSE or an EDIT work surface or if a program or procedure will be called. If BROWSE is specified, the related data will appear within the named window in a BROWSE work surface. If EDIT is used the first related record will be shown in an EDIT work surface. If DO is specified, the named program or procedure will be called.

The BROWSE or EDIT command may be customized with any qualifier, by using the <qualifier> option. For example, when specifying BROWSE, you might want to limit the number of fields with the FIELDS qualifier of the BROWSE command.

If the SET WINDOW OF RELATION command is issued without any qualifiers it will deactivate the related window. To re-activate the related window, issue another SET WINDOW OF RELATION command.

Example

```
set view to demo  
set window of relation in accounts to brelation;  
    command browse  
set window of relation in product to bedit;
```


command edit
set window of browse to rbrowse
browse noclear

Products

Recital Terminal Developer

SET WP

Class

Recital Terminal Developer Environment

Purpose

Specify the name of the word processor to be used when editing memos

Syntax

SET WP TO <expC>

See Also

SET MEMOWIDTH, MEMLINES(), MEMOLINE(), MEMOREAD(), MEMOWRITE(), MLCOUNT(), MLINE(), TEXTEDIT()

Description

The SET WP TO <expC> command is used to specify the name of an external word processing program that will be used to edit memo fields. All memos whether created by an external word processor or by the Recital default editor will be saved in ASCII format. The MEMOEDIT() function is not affected by this command.

NOTE: The external word processor will only be activated if MEMOWINDOW is set OFF.

Example

```
set wp to "wm"  
set memowindow off
```

Products

Recital Terminal Developer

SET WRAP

Class

Screen Environment

Purpose

Control cursor movement between menu options

Syntax

SET WRAP ON | OFF | (<expL>)

See Also

SET MCONFIRM

Description

The SET WRAP command controls cursor movement between the first and last options in menus. If WRAP is on and the cursor is positioned on the last menu option, pressing the [CURSOR DOWN] key moves to the first option. Similarly when on the first menu option, pressing the [CURSOR UP] key moves to the last one. By default, WRAP is ON.

Example

```
set wrap on
```

Products

Recital Terminal Developer

SET XMLFORMAT

Class

SQL Applications

Purpose

Specify the default format for XML files created by SELECT...SAVE AS XML

Syntax

SET XMLFORMAT TO <RECITAL | ADO>

See Also

SELECT

Description

The SET XMLFORMAT TO <RECITAL | ADO> command allows you to specify the default format for XML files created by SELECT...SAVE AS XML. The XMLFORMAT can be either RECITAL or ADO (Microsoft® ActiveX® Data Objects). Any XML files created in the ADO format can be loaded with the Open method of an ADO Recordset object.

The default XMLFORMAT setting is ADO. The default XMLFORMAT setting can also be overridden using the FORMAT clause on the SELECT statement.

Example

```
set xmlformat to ADO
EXEC SQL
SELECT * FROM example
SAVE AS XML example;
// In Visual Basic the file can then be loaded:
// Set adoPrimaryRS = New Recordset
// adoPrimaryRS.Open "example.xml"
```

Products

Recital Database Server, Recital Mirage Server, Recital Terminal Developer