

# Lotus Spreadsheet for DeskMate

## Quick Reference

### Key Summary

Key Name	Action
HELP (F1 or CTRL-F1)	Provides context-sensitive information on Lotus-DM commands and error messages.
COMPOSE (ALT-F1)	Creates international characters when used in combination with two other keys.
EDIT (CTRL-F2)	Places selected entry on the edit panel for editing.
NAME (CTRL-F3)	Displays a list of the current range names, from which you can select a name to use in a field.
ABS (CTRL-F4)	Cycles a cell address through relative, absolute, and mixed in POINT and EDIT modes.
GOTO (CTRL-F5)	Moves the cell pointer directly to the cell or to the top left cell in a named range you specify.
QUERY (CTRL-F7)	Repeats the last Data Query you used during the current session.
TABLE (CTRL-F8)	Repeats the last Data Table you used during the current session.
CALC (CTRL-F9)	Recalculates all formulas in a worksheet in READY mode; converts formula into its current value in VALUE and EDIT modes.
GRAPH VIEW (CTRL-F10)	Displays the current graph with its current settings on the screen.
SHIFT-ENTER	Activates POINT mode and anchors the current cell.

Neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of Lotus Development Corporation, except in the manner described in the documentation.

© Copyright 1989 Lotus Development Corporation  
55 Cambridge Parkway  
Cambridge, MA 02142

All rights reserved. Printed in the United States.

@SIN(*x*) calculates the sine of angle *x*.

@SLN(*cost,salvage,life*) calculates the straight-line depreciation allowance of an asset for one period.

@SQRT(*x*) calculates the positive square root of *x*.

@STD(*list*) calculates the population standard deviation of the values in *list*.

@STRING(*x,n*) converts the value (*x*) into a string with *n* decimal places.

@SUM(*list*) sums the values in *list*.

@SYD(*cost,salvage,life,period*) calculates the sum-of-the-years' digits depreciation allowance of an asset for a specified *period*.

@TAN(*x*) calculates the tangent of angle *x*.

@TERM(*payments,interest,future-value*) calculates the number of payment periods in the term of an investment necessary to accumulate a *future-value*, assuming *payments* of equal value, when the investment earns a periodic *interest* rate.

@TIME(*hour,minutes,seconds*) returns the time number for *hour*, *minutes*, and *seconds*.

@TIMEVALUE(*string*) returns the time number for a *string* that represents a time.

@TRIM(*string*) returns *string* with no leading, trailing, or consecutive spaces.

@TRUE returns the logical value 1 (true).

@UPPER(*string*) converts all the letters in *string* to uppercase.

@VALUE(*string*) converts a number entered as a *string* into its corresponding numeric value.

@VAR(*list*) calculates the population variance of the values in *list*.

@VLOOKUP(*x,range,column-offset*) returns the contents of a cell in a specified column of a vertical lookup table.

@YEAR(*date-number*) returns the year, an integer from 0 (1900) to 199 (2099), in *date-number*.

@MID(*string,start-number,n*) returns *n* characters in *string*, beginning with the character at *start-number*.

@MIN(*list*) finds the smallest value in *list*.

@MINUTE(*time-number*) returns the minutes, an integer from 0 to 59, in *time-number*.

@MOD(*x,y*) calculates the remainder (modulus) of *x/y*.

@MONTH(*date-number*) returns the month, an integer from 1 to 12, in *date-number*.

@N(*range*) returns a value for the first cell in *range*. If the cell contains a label, @N returns the value 0.

@NA returns the value NA (not available).

@NOW returns the date and time number that corresponds to the current date and time on the computer's clock.

@NPV(*interest,range*) calculates the net present value of a series of future cash flows discounted at a fixed, periodic *interest* rate.

@PI returns the value  $\pi$  (calculated at 3.1415926536).

@PMT(*principal,interest,term*) calculates the amount of the periodic payment needed to pay off a loan, given a specified periodic *interest* rate and number of payment periods.

@PROPER(*string*) converts the first letter in each word in *string* to uppercase and the remaining letters to lowercase.

@PV(*payments,interest,term*) calculates the present value of an investment.

@RAND generates a random number between 0 and 1.

@RATE(*future-value,present-value,term*) calculates the periodic interest rate necessary for an investment (*present-value*) to grow to a *future-value* over the number of compounding periods in *term*.

@REPEAT(*string,n*) duplicates *string* *n* times.

@REPLACE(*original-string,start-number,n,new-string*) replaces *n* characters in *original-string*, beginning at *start-number*, with *new-string*.

@RIGHT(*string,n*) returns last *n* characters in *string*.

@ROUND(*x,n*) rounds the number *x* to *n* places.

@ROWS(*range*) counts the rows in *range*.

@S(*range*) returns the entry in the first cell in *range* as a label. If the cell contains a label, @S returns that label; if the cell contains a value, @S returns an empty string.

@SECOND(*time-number*) returns the seconds, an integer from 0 to 59, in *time-number*.

@FALSE returns the logical value 0 (false).

@FIND(*search-string,string,start-number*) calculates the position in *string*, beginning with *start-number*, at which @FIND finds the first occurrence of *search-string*.

@FV(*payments,interest,term*) calculates the future value of an investment, based on a series of equal *payments*, earning a periodic *interest* rate, over the number of payment periods in *term*.

@HLOOKUP(*x,range,row-offset*) returns the contents of a cell in a specified row of a horizontal lookup table.

@HOUR(*time-number*) returns the hour, an integer from 0 to 23, in *time-number*.

@IF(*condition,x,y*) evaluates *condition* as true or false and takes one of two actions: if *condition* is true, @IF returns *x*; if *condition* is false, @IF returns *y*.

@INDEX(*range,column-offset,row-offset*) finds the value in the cell located at a specified *column-offset* and *row-offset* of *range*.

@INT(*x*) returns the integer portion of *x*.

@IRR(*guess,range*) calculates the internal rate of return expected from a series of cash flows generated by an investment.

@ISERR(*x*) tests *x* for the value ERR. If *x* is the value ERR, @ISERR returns 1 (true); if *x* is not the value ERR, @ISERR returns 0 (false).

@ISNA(*x*) tests *x* for the value NA. If *x* is the value NA, @ISNA returns 1 (true); if *x* is not the value NA, @ISNA returns 0 (false).

@ISNUMBER(*x*) tests *x* for a value. If *x* is a number, the value ERR, the value NA, or a blank cell, @ISNUMBER returns 1 (true); if *x* is a string, @ISNUMBER returns 0 (false).

@ISSTRING(*x*) tests *x* for a string. If *x* is a literal string or reference to a cell that contains a label, @ISSTRING returns 1 (true); if *x* is a number, the value ERR, the value NA, or a blank cell, @ISSTRING returns 0 (false).

@LEFT(*string,n*) returns the first *n* characters in *string*.

@LENGTH(*string*) counts the characters in *string*.

@LN(*x*) calculates the natural logarithm (base *e*) of *x*.

@LOG(*x*) calculates the common logarithm (base 10) of *x*.

@LOWER(*string*) converts all the letters in *string* to lowercase.

@MAX(*list*) finds the largest value in *list*.

## Label Prefixes

Prefix	Alignment
' (apostrophe)	Left
" (double quote)	Right
^ (caret)	Center
\ (backslash)	Repeating

## Command Accelerators

Command	Accelerator
File Save	CTRL-S
File Print	CTRL-P
Edit Copy Range	CTRL-C
Edit Move Range	CTRL-M
Edit Cut	SHIFT-DEL
Edit Copy	CTRL-INS
Edit Paste	SHIFT-INS
Edit Clear	DEL
Worksheet Column Width	CTRL-W
Worksheet Suppress Zero	CTRL-Z
Worksheet Grid	CTRL-G
Range Format	CTRL-F
Range Name	CTRL-N
Graph View	CTRL-F10
Graph Ranges	CTRL-R
Graph Name	CTRL-E
Graph Type	CTRL-T
Data Fill	CTRL-D
Exit	ESC

## Pointer-Movement Keys

Key Name	Action
↑ (UP)	Moves cell pointer up one cell in READY and POINT modes; moves cell pointer up one item in HELP mode; completes entry and moves cell pointer up one cell.
↓ (DOWN)	Moves cell pointer down one cell in READY and POINT modes; moves down one item in HELP mode; completes entry and moves cell pointer down one cell in EDIT mode.
← (LEFT)	Moves cell pointer left one cell in READY and POINT modes; moves left one item in HELP mode.
→ (RIGHT)	Moves cell pointer right one cell in READY and POINT modes; moves right one item in HELP mode.
CTRL - ← (BIG LEFT)	Moves pointer one screen to the left in READY mode; moves cursor to the beginning of the edit panel entry in EDIT mode.
CTRL - → (BIG RIGHT)	Moves pointer one screen to the right in READY mode; moves cursor to the end of the edit panel entry in EDIT mode.
PAGE DOWN	Moves cell pointer down one page in READY and POINT modes; completes an entry and moves cell pointer down one screen in EDIT mode.
PAGE UP	Moves cell pointer up one page in READY and POINT modes; completes an entry and moves cell pointer up one screen in EDIT mode.
HOME	Moves cell pointer to cell A1 in READY and POINT modes; moves to first character of edit panel in EDIT mode.
END	Moves cell pointer to last item in HELP mode; moves to last character in EDIT mode.
END DOWN	Moves cell pointer down to next cell where blank and filled cells meet, in READY and POINT modes.
END HOME	Moves cell pointer to bottom right corner of active area, in READY and POINT modes.
END LEFT	Moves cell pointer left to next cell where blank and filled cells meet, in READY and POINT modes.
END RIGHT	Moves cell pointer right to next cell where blank and filled cells meet, in READY and POINT modes.
END UP	Moves cell pointer up to next cell where blank and filled cells meet, in READY and POINT modes.
GOTO (CTRL-F5)	Moves cell pointer directly to the cell or to the top left cell in a named range you specify.

## Order of Precedence for Operators

Operator	Operation	Precedence Number
^	Exponentiation	1 (highest)
- +	Identification of value as negative or positive	2
* /	Multiplication and division	3
+ -	Addition and subtraction	4
= <>	Equal-to and not-equal-to tests	5
< >	Less-than and greater-than tests	5
<=	Less-than-or-equal-to test	5
>=	Greater-than-or-equal-to test	5
#NOT#	Logical-NOT test	6
#AND# #OR#	Logical-AND and logical-OR tests	7
&	String concatenation	7 (lowest)

## @Functions

**@(cell address)** returns the contents of the cell whose name or address is specified in *cell address*.

**@ABS(x)** calculates the absolute value of *x*.

**@ACOS(x)** calculates the arc cosine of *x*.

**@ASIN(x)** calculates the arc sine of *x*.

**@ATAN(x)** calculates the arc tangent of *x*.

**@ATAN2(x,y)** calculates the four-quadrant arc tangent of *y/x*.

**@AVG(list)** averages the values in *list*.

**@CELL(attribute,range)** returns information about an *attribute* for the first cell in *range*.

**@CELLPOINTER(attribute)** returns information about an *attribute* for the current cell.

**@CHAR(x)** returns the LICS character for the corresponding *x* value.

**@CHOOSE(offset,list)** returns the value or string in a list specified by *offset*.

**@CODE(string)** returns the LICS value that corresponds to the first character in *string*.

**@COLS(range)** counts the columns in *range*.

**@COS(x)** calculates the cosine of angle *x*.

**@COUNT(list)** counts the nonblank cells in a *list* of ranges.

**@CTERM(interest,future-value,present-value)** calculates the number of compounding periods it takes for an investment (*present-value*) to grow to a *future-value*, earning a fixed *interest* rate per compounding period.

**@DATE(year,month,day)** returns the date number for *year*, *month*, and *day*.

**@DATEVALUE(string)** returns the date number for a *string* that represents a date.

**@DAVG(input,offset,criteria)** averages the values in the offset column of a database table, based on certain criteria.

**@DAY(date-number)** returns the day of the month, an integer from 1 to 31, in *date-number*.

**@DCOUNT(input,offset,criteria)** counts the nonblank cells in the offset column of a database table, based on certain criteria.

**@DDB(cost,salvage,life,period)** calculates the depreciation allowance of an asset for a specified *period*, using the double-declining balance method.

**@DMAX(input,offset,criteria)** finds the largest value in the offset column of a database table, based on certain criteria.

**@DMIN(input,offset,criteria)** finds the smallest value in the offset column of a database table, based on certain criteria.

**@DSTD(input,offset,criteria)** calculates the population standard deviation of the values in the offset column of a database table, based on certain criteria.

**@DSUM(input,offset,criteria)** sums the values in the offset column of a database table, based on certain criteria.

**@DVAR(input,offset,criteria)** calculates the population variance of the values in the offset column of a database, based on certain criteria.

**@ERR** returns the value ERR (error).

**@EXACT(string1,string2)** returns 1 (true) if *string1* and *string2* are the same or 0 (false) if they are not the same.

**@EXP(x)** calculates the value of *e* (2.71828...) raised to the *x*th power.