NAME

xcalc – scientific calculator for X

SYNOPSIS

xcalc [-stipple] [-rpn] [-*toolkitoption...*]

DESCRIPTION

xcalc is a scientific calculator desktop accessory that can emulate a TI-30 or an HP-10C.

OPTIONS

xcalc accepts all of the standard toolkit command line options along with two additional options:

–stipple
This option indicates that the background of the calculator should be drawn using a stipple of the foreground and background colors. On monochrome displays improves the appearance.

–rpn This option indicates that Reverse Polish Notation should be used. In this mode the calculator will look and behave like an HP-10C. Without this flag, it will emulate a TI-30.

OPERATION

Pointer Usage: Operations may be performed with pointer button 1, or in some cases, with the keyboard. Many common calculator operations have keyboard accelerators. To quit, press pointer button 3 on the AC key of the TI calculator, or the ON key of the HP calculator.

Calculator Key Usage (TI mode): The numbered keys, the +/- key, and the +, -, ∗, /, and = keys all do exactly what you would expect them to. It should be noted that the operators obey the standard rules of precedence. Thus, entering "3+4∗5=" results in "23", not "35". The parentheses can be used to override this. For example, "(1+2+3)∗(4+5+6)=" results in "6∗15=90".

The entire number in the calculator display can be selected, in order to paste the result of a calculation into text.

The action procedures associated with each function are given below. These are useful if you are interested in defining a custom calculator. The action used for all digit keys is **digit(***n*), where *n* is the corresponding digit, 0..9.

1/x Replaces the number in the display with its reciprocal. The corresponding action procedure is **reciprocal()**.

x^2 Squares the number in the display. The corresponding action procedure is **square()**.

SQRT Takes the square root of the number in the display. The corresponding action procedure is **squareRoot()**.

CE/C When pressed once, clears the number in the display without clearing the state of the machine. Allows you to re-enter a number if you make a mistake. Pressing it twice clears the state, also. The corresponding action procedure for TI mode is **clear()**.

AC Clears the display, the state, and the memory. Pressing it with the third pointer button turns off the calculator, in that it exits the program. The action procedure to clear the state is **off()**; to quit, **quit()**.

INV Invert function. See the individual function keys for details. The corresponding action procedure is **inverse()**.

sin Computes the sine of the number in the display, as interpreted by the current DRG mode (see DRG, below). If inverted, it computes the arcsine. The corresponding action procedure is **sine()**.

cos Computes the cosine, or arccosine when inverted. The corresponding action procedure is **cosine()**.

tan Computes the tangent, or arctangent when inverted. The corresponding action procedure is **tangent()**.

DRG Changes the DRG mode, as indicated by 'DEG', 'RAD', or 'GRAD' at the bottom of of the calculator ''liquid crystal'' display. When in 'DEG' mode, numbers in the display are taken as being degrees. In 'RAD' mode, numbers are in radians, and in 'GRAD' mode, numbers are in grads. When inverted, the DRG key has a feature of converting degrees to radians to grads and vice-versa. Example: put the calculator into 'DEG' mode, and enter "45 INV DRG". The display should now show something along the lines of ".785398", which is 45 degrees converted to radians. The corresponding action procedure is **degree()**.

e The constant 'e'. (2.7182818...). The corresponding action procedure is e().

EE Used for entering exponential numbers. For example, to get "-2.3E-4" you'd enter "2 . 3 +/- EE 4 +/-". The corresponding action procedure is **scientific()**.

log Calculates the log (base 10) of the number in the display. When inverted, it raises "10.0" to the number in the display. For example, entering "3 INV log" should result in "1000". The corresponding action procedure is **logarithm()**.

ln Calculates the log (base e) of the number in the display. When inverted, it raises "e" to the number in the display. For example, entering "e ln" should result in "1". The corresponding action procedure is naturalLog().

y^x Raises the number on the left to the power of the number on the right. For example "2 y^x 3 =" results in "8", which is 2^3. For a further example, "(1+2+3) y^x (1+2) =" equals "6 y^x 3" which equals "216". The corresponding action procedure is **power()**.

PI The constant 'pi'. (3.1415927....) The corresponding action procedure is **pi()**.

x! Computes the factorial of the number in the display. The number in the display must be an integer in the range 0-500, though, depending on your math library, it might overflow long before that. The corresponding action procedure is **factorial()**.

(Left parenthesis. The corresponding action procedure for TI calculators is **leftParen()**.

) Right parenthesis. The corresponding action procedure for TI calculators is **rightParen()**.

/ Division. The corresponding action procedure is **divide()**.

* Multiplication. The corresponding action procedure is **multiply()**.

- Subtraction. The corresponding action procedure is **subtract()**.

+ Addition. The corresponding action procedure is **add()**.

= Perform calculation. The TI-specific action procedure is **equal()**.

STO Copies the number in the display to the memory location. The corresponding action procedure is **store()**.

RCL Copies the number from the memory location to the display. The corresponding action procedure is **recall()**.

SUM Adds the number in the display to the number in the memory location. The corresponding action procedure is **sum()**.

EXC Swaps the number in the display with the number in the memory location. The corresponding action procedure for the TI calculator is **exchange()**.

+/- Negate; change sign. The corresponding action procedure is **negate()**.

. Decimal point. The action procedure is **decimal()**.

Calculator Key Usage (RPN mode): The number keys, CHS (change sign), +, -, *, /, and ENTR keys all do exactly what you would expect them to do. Many of the remaining keys are the same as in TI mode. The differences are detailed below. The action procedure for the ENTR key is enter().

<- This is a backspace key that can be used if you make a mistake while entering a number. It will erase digits from the display. (See BUGS). Inverse backspace will clear the X register. The corresponding action procedure is **back()**.

ON Clears the display, the state, and the memory. Pressing it with the third pointer button turns off the calculator, in that it exits the program. To clear state, the action procedure is **off**; to quit, **quit()**.

INV Inverts the meaning of the function keys. This would be the *f* key on an HP calculator, but *xcalc* does not display multiple legends on each key. See the individual function keys for details.

10^x Raises "10.0" to the number in the top of the stack. When inverted, it calculates the log (base 10) of the number in the display. The corresponding action procedure is **tenpower()**.

e^x Raises "e" to the number in the top of the stack. When inverted, it calculates the log (base e) of the number in the display. The action procedure is **epower()**.

STO Copies the number in the top of the stack to a memory location. There are 10 memory locations. The desired memory is specified by following this key with a digit key.

RCL Pushes the number from the specified memory location onto the stack.

SUM Adds the number on top of the stack to the number in the specified memory location.

x:y Exchanges the numbers in the top two stack positions, the X and Y registers. The corresponding action procedure is **XexchangeY()**.

R v Rolls the stack downward. When inverted, it rolls the stack upward. The corresponding action procedure is **roll()**.

blank These keys were used for programming functions on the HP-10C. Their functionality has not been duplicated in *xcalc*.

Finally, there are two additional action procedures: **bell()**, which rings the bell; and **selection()**, which performs a cut on the entire number in the calculator's ''liquid crystal'' display.

ACCELERATORS

Accelerators are shortcuts for entering commands. *xcalc* provides some sample keyboard accelerators; also users can customize accelerators. The numeric keypad accelerators provided by *xcalc* should be intuitively correct. The accelerators defined by *xcalc* on the main keyboard are given below:

TI Key	HP Key	Keyboard Accelerator	TI Function	HP Function
SQRT	SQRT	r	squareRoot()	squareRoot()
AC	ON	space	clear()	clear()
AC	<-	Delete	clear()	back()
AC	<-	Backspace	clear()	back()
AC	<-	Control-H	clear()	back()
AC		Clear	clear()	
AC	ON	q	quit()	quit()
AC	ON	Control-C	quit()	quit()
INV	i	i	inverse()	inverse()
sin	s	s	sine()	sine()
cos	c	c	cosine()	cosine()
tan	t	t	tangent()	tangent()
DRG	DRG	d	degree()	degree()
e		e	e()	
ln	ln	l	naturalLog()	naturalLog()
y^x	y^x	^	power()	power()
PI	PI	p	pi()	pi()
x!	x!	!	factorial()	factorial()
(		(	leftParen()	
)		)	rightParen()	
/	/	/	divide()	divide()
*	*	*	multiply()	multiply()
-	-	-	subtract()	subtract()

+	+	+	add()	add()
=		=	equal()	
0..9	0..9	0..9	digit()	digit()
.	.	.	decimal()	decimal()
+/-	CHS	n	negate()	negate()
	x:y	x		XexchangeY()
	ENTR	Return		enter()
	ENTR	Linefeed		enter()

CUSTOMIZATION

The application class name is XCalc.

xcalc has an enormous application defaults file which specifies the position, label, and function of each key on the calculator. It also gives translations to serve as keyboard accelerators. Because these resources are not specified in the source code, you can create a customized calculator by writing a private application defaults file, using the Athena Command and Form widget resources to specify the size and position of buttons, the label for each button, and the function of each button.

The foreground and background colors of each calculator key can be individually specified. For the TI calculator, a classical color resource specification might be:

```
XCalc.ti.Command.background: gray50
XCalc.ti.Command.foreground:  white
```

For each of buttons 20, 25, 30, 35, and 40, specify:

```
XCalc.ti.button20.background:   black
XCalc.ti.button20.foreground:   white
```

For each of buttons 22, 23, 24, 27, 28, 29, 32, 33, 34, 37, 38, and 39:

```
XCalc.ti.button22.background:   white
XCalc.ti.button22.foreground:   black
```

WIDGET HIERARCHY

In order to specify resources, it is useful to know the hierarchy of the widgets which compose xcalc. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
XCalc xcalc
        Form  ti  or  hp    (the name depends on the mode)
                Form  bevel
                        Form  screen
                                Label  M
                                Toggle  LCD
                                Label  INV
                                Label  DEG
                                Label  RAD
```

```
                        Label  GRAD
                        Label  P
            Command  button1
            Command  button2
            Command  button3
```
and so on, ...
```
            Command  button38
            Command  button39
            Command  button40
```

APPLICATION RESOURCES

rpn (Class **Rpn**)
Specifies that the rpn mode should be used. The default is TI mode.

stipple (Class **Stipple**)
Indicates that the background should be stippled. The default is ''on'' for monochrome displays, and ''off'' for color displays.

cursor (Class **Cursor**)
The name of the symbol used to represent the pointer. The default is ''hand2''.

COLORS

If you would like xcalc to use its ti colors, include the following in the #ifdef COLOR section of the file you read with xrdb:

```
*customization:            -color
```

This will cause xcalc to pick up the colors in the app-defaults color customization file: /usr/openwin/lib/app-defaults/XCalc-color.

SEE ALSO

X11(7), **xrdb**(1), the Athena Widget Set

BUGS

HP mode: A bug report claims that the sequence of keys 5, ENTER, **<-** should clear the display, but it doesn't.

COPYRIGHT

AUTHORS

John Bradley, University of Pennsylvania
Mark Rosenstein, MIT Project Athena
Donna Converse, MIT X Consortium