MOTIVATION

The following is a very brief description of all the basic primitives currently available in MUDDLE. These descriptions are in no way intended to be a primer on MUDDLE programming. Neither are they to be considered a definition of the effects or values produced by the primitives. I have tried to be as complete and as accurate as is possible in a single statement description. However, because of the complexity of most primitives, many important defaults and restrictions have been omitted. It is hoped that by using this manual the user can be aware of what facilities exist and then look elsewhere for a precise definition of those primitives which he believes might be useful.

DESCRIPTION

A given description contains three pieces of information about each primitive: its name, its description, and the number of arguments it takes. The name is just the text that is used to refer to each primitive. Also indicated is whether the primitive evaluates its arguments (SUBR) or doesn't evaluate its arguments (FSUBR). Even though all primitives return a value, some
descriptions only mention the side effects produced by a primitive. These primitives are most often used for this effect rather than the value, so the value is omitted. The third field indicates how many arguments the primitive can and usually must be supplied.
<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DESCRIPTION</th>
<th>NUMBER OF ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>* SUBR</td>
<td>Arithmetic: multiplication</td>
<td>any</td>
</tr>
<tr>
<td>+ SUBR</td>
<td>Arithmetic: addition</td>
<td>any</td>
</tr>
<tr>
<td>- SUBR</td>
<td>Arithmetic: subtraction</td>
<td>any</td>
</tr>
<tr>
<td>/ SUBR</td>
<td>Arithmetic: division</td>
<td>any</td>
</tr>
<tr>
<td>0? SUBR</td>
<td>Predicate: equality to number zero</td>
<td>1</td>
</tr>
<tr>
<td>1? SUBR</td>
<td>Predicate: equality to number one</td>
<td>1</td>
</tr>
<tr>
<td>==? SUBR</td>
<td>Predicate: &quot;exact&quot; equality (sharing)</td>
<td>2</td>
</tr>
<tr>
<td>=? SUBR</td>
<td>Predicate: &quot;structural&quot; equality</td>
<td>2</td>
</tr>
<tr>
<td>ABS SUBR</td>
<td>Arithmetic: absolute value</td>
<td>1</td>
</tr>
<tr>
<td>AGAIN SUBR</td>
<td>Restarts a given activation block</td>
<td>0-1</td>
</tr>
<tr>
<td>ALLTYPES SUBR</td>
<td>Returns a vector of all currently known data types</td>
<td>0</td>
</tr>
<tr>
<td>AND FSUBR</td>
<td>Logical: &quot;and&quot; of truthvalues</td>
<td>any</td>
</tr>
<tr>
<td>ARGS SUBR</td>
<td>Returns arguments of a given FRAME</td>
<td>1</td>
</tr>
<tr>
<td>ASCII SUBR</td>
<td>Returns character with a given &quot;ASCII&quot; code</td>
<td>1</td>
</tr>
<tr>
<td>ASSIGNED? SUBR</td>
<td>Predicate: is an ATOM locally assigned</td>
<td>1</td>
</tr>
<tr>
<td>AT SUBR</td>
<td>Returns a LOCATIVE to the nth element of a structure</td>
<td>1-2</td>
</tr>
<tr>
<td>ATAN SUBR</td>
<td>Arithmetic: arc tangent</td>
<td>1</td>
</tr>
<tr>
<td>ATOM SUBR</td>
<td>Creates an ATOM with a given name</td>
<td>1</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Argument(s)</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BACK SUBR</td>
<td>Replaces some items removed from a non-LIST structure by RESTing</td>
<td>1-2</td>
</tr>
<tr>
<td>BITS SUBR</td>
<td>Returns the specification of a bit field in a WORD</td>
<td>0-2</td>
</tr>
<tr>
<td>BLOCK SUBR</td>
<td>Creates a new path of OBLISTs for READING</td>
<td>1</td>
</tr>
<tr>
<td>BOUND? SUBR</td>
<td>Predicate: is an ATOM locally bound</td>
<td>1</td>
</tr>
<tr>
<td>CHANLIST SUBR</td>
<td>Returns a LIST of currently open CHANNELS for I/O</td>
<td>0</td>
</tr>
<tr>
<td>CHANNEL SUBR</td>
<td>Creates a CHANNEL for I/O</td>
<td>0-5</td>
</tr>
<tr>
<td>CHTYPE SUBR</td>
<td>Changes the data type of an item</td>
<td>2</td>
</tr>
<tr>
<td>CHUTYPE SUBR</td>
<td>Changes the data type of the elements of a UVECTOR</td>
<td>2</td>
</tr>
<tr>
<td>CLOSE SUBR</td>
<td>Closes a CHANNEL for I/O</td>
<td>1</td>
</tr>
<tr>
<td>CLOSURE SUBR</td>
<td>Binds the free variables of a FUNCTION to current values</td>
<td>1-more</td>
</tr>
<tr>
<td>COND FSUBR</td>
<td>Conditional evaluation of expressions</td>
<td>any</td>
</tr>
<tr>
<td>CONS SUBR</td>
<td>Adds an item to the front of a LIST</td>
<td>2</td>
</tr>
<tr>
<td>COS SUBR</td>
<td>Arithmetic: cosine</td>
<td>1</td>
</tr>
<tr>
<td>CREATE SUBR</td>
<td>Creates a new PROCESS</td>
<td>1</td>
</tr>
<tr>
<td>ECHOPAIR SUBR</td>
<td>Sets up CHANNELs for echoing characters on rubout</td>
<td>2</td>
</tr>
<tr>
<td>EMPTY? SUBR</td>
<td>Predicate: does a structure have zero elements</td>
<td>1</td>
</tr>
<tr>
<td>ENDBLOCK SUBR</td>
<td>Restores previous path of OBLISTs before last call to BLOCK</td>
<td>0</td>
</tr>
<tr>
<td>ERRET SUBR</td>
<td>Proceeds evaluation from the last ERROR or LISTEN</td>
<td>0-2</td>
</tr>
</tbody>
</table>
ERROR SUBR  Stops and informs user of an error

ERRORS SUBR  Returns the OBLIST where error messages are located 0

EVAL SUBR  Evaluates an expression in a given environment 1-2

EXIT SUBR  Leaves an activation block with a given value 2

EXP SUBR  Arithmetic: exponentiation to the base "e" 1

FAIL SUBR  PLANNER primitive 0-2

FAILPOINT FSUBR  PLANNER primitive 1

FALSE SUBR  Predicate: returns truthvalue of "false" 0-1

FINIALIZE SUBR  PLANNER primitive 1

FIX SUBR  Arithmetic: returns FIX value of a number 1

FLATSIZE SUBR  Returns number of characters needed to print an item 2

FLOAD SUBR  Reads and evaluates all items of a file 0-5

FLOAT SUBR  Arithmetic: returns FLOAT value of a number 1

FRAME SUBR  Returns a previous FRAME 0-1

FUNCT SUBR  Returns function name of a given FRAME 1

FUNCTION FSUBR  Creates a FUNCTION 2-more

G? SUBR  Predicate: is first argument numerically greater than second 2

GASSIGNED? SUBR  Predicate: is an ATOM globally assigned 1

GET SUBR  Returns a given property associated with an item 2-3
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETBITS SUBR</td>
<td>Extracts a specified bit field from a WORD</td>
<td>2</td>
</tr>
<tr>
<td>GETINT SUBR</td>
<td>Returns the number of the most recent interrupt</td>
<td>0</td>
</tr>
<tr>
<td>GETPROP SUBR</td>
<td>A more general version of GET</td>
<td>2-3</td>
</tr>
<tr>
<td>GLOC SUBR</td>
<td>Returns a LOCATIVE to the global value cell of an ATOM</td>
<td>1</td>
</tr>
<tr>
<td>GO SUBR</td>
<td>Goes to a tag and continues evaluation from there</td>
<td>1</td>
</tr>
<tr>
<td>GROW SUBR</td>
<td>Extends the bounds of a VECTOR or UVVECTOR</td>
<td>3</td>
</tr>
<tr>
<td>GVAL SUBR</td>
<td>Returns the global value of an ATOM</td>
<td>1</td>
</tr>
<tr>
<td>I LIST SUBR</td>
<td>Creates a LIST with implicit elements</td>
<td>1-2</td>
</tr>
<tr>
<td>IN SUBR</td>
<td>Returns the item pointed to by a LOCATIVE</td>
<td>1</td>
</tr>
<tr>
<td>INSERT SUBR</td>
<td>Adds an ATOM to an OBLIST</td>
<td>2</td>
</tr>
<tr>
<td>INTCHAN SUBR</td>
<td>Returns the number of the most recent channel to be interrupted</td>
<td>0</td>
</tr>
<tr>
<td>INTCHAR SUBR</td>
<td>Returns an interrupt level CHARACTER from a CHANNEL</td>
<td>1</td>
</tr>
<tr>
<td>INTERN SUBR</td>
<td>Inserts an ATOM IN a given OBLIST</td>
<td>2</td>
</tr>
<tr>
<td>INTERRUPTS SUBR</td>
<td>Returns the OBLIST on which interrupt routines are kept</td>
<td>0</td>
</tr>
<tr>
<td>ISTRING SUBR</td>
<td>Creates a STRING with implicit elements</td>
<td>1-2</td>
</tr>
<tr>
<td>IUVECTOR SUBR</td>
<td>Creates a UVVECTOR with implicit elements</td>
<td>1-2</td>
</tr>
<tr>
<td>IVECTOR SUBR</td>
<td>Creates a VECTOR with implicit elements</td>
<td>1-2</td>
</tr>
<tr>
<td>L? SUBR</td>
<td>Predicate: is first argument numerically less than the second</td>
<td>2</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td>Arguments</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>LENGTH SUBR</td>
<td>Returns the number of elements in a structure</td>
<td>1</td>
</tr>
<tr>
<td>LIST SUBR</td>
<td>Creates a LIST with explicit elements</td>
<td>any</td>
</tr>
<tr>
<td>LISTEN SUBR</td>
<td>Stops and informs user that you are waiting</td>
<td>any</td>
</tr>
<tr>
<td>LLOC SUBR</td>
<td>Returns a LOCATIVE to the local value cell of an ATOM</td>
<td>1</td>
</tr>
<tr>
<td>LOAD SUBR</td>
<td>Reads and evaluates all items from a CHANNEL</td>
<td>1-2</td>
</tr>
<tr>
<td>LOG SUBR</td>
<td>Arithmetic: natural logarithm</td>
<td>1</td>
</tr>
<tr>
<td>LOOKUP SUBR</td>
<td>Returns an ATOM found on a given OBLIST</td>
<td>2</td>
</tr>
<tr>
<td>LVAL SUBR</td>
<td>Returns the local value of an ATOM</td>
<td>1</td>
</tr>
<tr>
<td>MAX SUBR</td>
<td>Arithmetic: maximum argument</td>
<td>any</td>
</tr>
<tr>
<td>MEMBER SUBR</td>
<td>Predicate: is item =? to some element of a structure</td>
<td>2</td>
</tr>
<tr>
<td>MEMQ SUBR</td>
<td>Predicate: is item ==? to some element of a structure</td>
<td>2</td>
</tr>
<tr>
<td>MIN SUBR</td>
<td>Arithmetic: minimum argument</td>
<td>any</td>
</tr>
<tr>
<td>MOBLIST SUBR</td>
<td>Creates an OBLIST</td>
<td>0-1</td>
</tr>
<tr>
<td>MOD SUBR</td>
<td>Arithmetic: numerical modulus or remainder</td>
<td>2-2</td>
</tr>
<tr>
<td>MONAD? SUBR</td>
<td>Predicate: is item unstructured or else EMPTY? structure</td>
<td>1</td>
</tr>
<tr>
<td>NEWTYPE SUBR</td>
<td>Defines a new data type</td>
<td>2</td>
</tr>
<tr>
<td>NEXTCHR SUBR</td>
<td>Returns the next CHARACTER from a CHANNEL</td>
<td>0-3</td>
</tr>
<tr>
<td>NOT SUBR</td>
<td>Logical: &quot;not&quot; of a truthvalue</td>
<td>1</td>
</tr>
<tr>
<td>NTH SUBR</td>
<td>Returns the nth element of a structure</td>
<td>1-2</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>OBLIST?</td>
<td>Predicate: is ATOM on an OBLIST</td>
<td></td>
</tr>
<tr>
<td>ONCHAR</td>
<td>Assigns an interrupt routine for a given CHANNEL</td>
<td></td>
</tr>
<tr>
<td>ONCLOCK</td>
<td>Assigns an interrupt routine for the slow clock break</td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td>Creates and opens a CHANNEL for I/O</td>
<td></td>
</tr>
<tr>
<td>OR FSUBR</td>
<td>Logical: &quot;or&quot; of truthvalues</td>
<td></td>
</tr>
<tr>
<td>PNAME</td>
<td>Returns a STRING which is the printing name of the ATOM</td>
<td></td>
</tr>
<tr>
<td>PRIMTYPE</td>
<td>Returns the primitive data type of an item</td>
<td></td>
</tr>
<tr>
<td>PRIN1</td>
<td>Prints an item on a CHANNEL without formatting</td>
<td></td>
</tr>
<tr>
<td>PRINC</td>
<td>Prints an item on a CHANNEL without formatting or indicators</td>
<td></td>
</tr>
<tr>
<td>PRINT</td>
<td>Prints an item on a CHANNEL</td>
<td></td>
</tr>
<tr>
<td>PROG FSUBR</td>
<td>Executes sequential expressions</td>
<td></td>
</tr>
<tr>
<td>PUT</td>
<td>Associates a property with an item</td>
<td></td>
</tr>
<tr>
<td>PUT1</td>
<td>Associates a property with an item</td>
<td></td>
</tr>
<tr>
<td>PUTBITS</td>
<td>Inserts a given bit field into a WORD</td>
<td></td>
</tr>
<tr>
<td>PUTN</td>
<td>Special version of PUT</td>
<td></td>
</tr>
<tr>
<td>PUTPROP</td>
<td>More general version of PUT</td>
<td></td>
</tr>
<tr>
<td>PUTREST</td>
<td>Replaces the REST of a LIST</td>
<td></td>
</tr>
<tr>
<td>QUITTER</td>
<td>Interrupt routine to handle !G quit feature</td>
<td></td>
</tr>
<tr>
<td>QUOTE FSUBR</td>
<td>Returns its argument unevaluated</td>
<td></td>
</tr>
<tr>
<td>RANDOM</td>
<td>Arithmetic: generate a uniform random fixed number</td>
<td></td>
</tr>
<tr>
<td>SUBR</td>
<td>Description</td>
<td>Arguments</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>READ SUBR</td>
<td>Reads one item from a CHANNEL 0-3</td>
<td>0-3</td>
</tr>
<tr>
<td>READCHR SUBR</td>
<td>Reads the next CHARACTER from a CHANNEL</td>
<td>0-3</td>
</tr>
<tr>
<td>REMOVE SUBR</td>
<td>Removes an ATOM from an OBLIST</td>
<td>2</td>
</tr>
<tr>
<td>REPEAT FSUBR</td>
<td>Executes repeatedly sequential expressions</td>
<td>2-more</td>
</tr>
<tr>
<td>RESET SUBR</td>
<td>Flushes the buffer of an I/O channel</td>
<td>1</td>
</tr>
<tr>
<td>REST SUBR</td>
<td>Removes the first n elements from a structure</td>
<td>1-2</td>
</tr>
<tr>
<td>RESTORE SUBR</td>
<td>PLANNER primitive</td>
<td>1-2</td>
</tr>
<tr>
<td>RESUME FSUBR</td>
<td>Restarts a PROCESS</td>
<td>1-2</td>
</tr>
<tr>
<td>RETURN SUBR</td>
<td>Leaves the most recent activation block with a given value</td>
<td>1</td>
</tr>
<tr>
<td>RSUBR SUBR</td>
<td>Generates a relocatable SUBR (used by the COMPILER)</td>
<td>1</td>
</tr>
<tr>
<td>ROOT SUBR</td>
<td>Returns the OBLIST containing primitives</td>
<td>0</td>
</tr>
<tr>
<td>SET SUBR</td>
<td>Changes the local value of an ATOM</td>
<td>2</td>
</tr>
<tr>
<td>SETG SUBR</td>
<td>Changes the global value of an ATOM</td>
<td>2</td>
</tr>
<tr>
<td>SETINT SUBR</td>
<td>Assigns an interrupt routine to a given interrupt number</td>
<td>2</td>
</tr>
<tr>
<td>SETLOC SUBR</td>
<td>Changes the contents pointed at by a LOCATIVE</td>
<td>2</td>
</tr>
<tr>
<td>SIN SUBR</td>
<td>Arithmetic: sine</td>
<td>1</td>
</tr>
<tr>
<td>SÖRT SUBR</td>
<td>Arithmetic: numerical sort of elements of a structure</td>
<td>1-2</td>
</tr>
<tr>
<td>SQRT SUBR</td>
<td>Arithmetic: square root</td>
<td>1</td>
</tr>
<tr>
<td>STACKFORM FSUBR</td>
<td>Applies a FUNCTION to arguments</td>
<td>3</td>
</tr>
</tbody>
</table>
STRING SUBR Creates a STRING with explicit any elements

TAG SUBR Creates a tag in an activation 1-2 block

TERPRI SUBR Prints a carriage return on 0-1 a CHANNEL

TIME SUBR Returns the system up in 30ths of a second 0

TOP SUBR Replaces all items removed from a non-LIST structure by 1 RESTing

TYPE SUBR Returns the data type of an item 1

UTYPE SUBR Returns the data type of the elements of a UVECTOR 1

UVECTOR SUBR Creates a UVECTOR with explicit any elements

VALUE SUBR Returns the local or else the global value of an ATOM 1

VECTOR SUBR Creates a VECTOR with explicit any elements