

● sys Module PREDICATION

A sys module is a set of the library module of the standard included in the system. When calling, it describes after ::sys.

::sys <args VAR>

ARG when starting the Descartes language is assigned to VAR.

::sys <DLIBPATH VAR>

The path DLIBPATH which the Descartes language uses is displayed on VAR.

::sys <mkpred ARG>

ARG is changed into PRED.

::sys <writeln LIST>

After outputting LIST, a new line is started.

::sys <write LIST>

::sys <w LIST>

Outputting LIST.

::sys <wnl>

A new-line is outputted.

::sys <wo NUMBER>

NUMBER is outputted by the octal number.

::sys <wx NUMBER>

NUMBER is outputted by a hexadecimal number.

::sys <wf NUMBER>

NUMBER is outputted as a floating point number.

::sys <wg NUMBER>

NUMBER is outputted in a floating point number and the optimal form.

::sys <wtab>

A tab is outputted.

::sys <fr VAR STRINGS WIDTH>

The result of having carried out flush right of the STRINGS by width is set as VAR.

::sys <fl VAR STRINGS WIDTH>

The result of having carried out flush left of the STRINGS by width is set as VAR.

::sys <isNil ARG>

::sys <isAtom ARG>

::sys <isList ARG>

::sys <isPred ARG>

::sys <isVar ARG>

::sys <isUndefVar ARG>

::sys <isFloat ARG>

::sys <isInteger ARG>

true will be returned, if ARG is judged and it corresponds. unknown is returned if it does not correspond.

::sys <isTrue PRED>

::sys <isFalse PRED>

::sys <isUnknown PRED>

The result of PRED of ARG is judged, and true will be returned if it corresponds. unknown is returned if it does not correspond.

::sys <max VAR LIST>

The greatest integer value of the element of a list is set as a variable.

::sys <min VAR LIST>

The minimum integer value of the element of a list is set as a variable.

::sys <maxf VAR LIST>

The greatest floating point number value of the element of a list is set as a variable.

::sys <minf VAR LIST>

The minimum floating point number value of the element of a list is set as a variable.

::sys <newobj NAME>

create a new NAME object.

::sys <delobj NAME>

delete NAME objects.

::sys <method NAME>

The method of a new name is generated

::sys <methoda NAME>

The method of a new name is generated from a head.

::sys <methodz NAME>

The method of a new name is generated at the end.

::sys <delmethod NAME>

The method of the specified name is deleted.

::sys <delmethoda NAME>

The method of the specified name is deleted from a head.

::sys <delmethodz NAME>

The method of the specified name is deleted from an end.

::sys <regex PATTERN STRINGS BEFORE-STRINGS MATCH-STRINGS AFTER-STRINGS>

The result of having applied the regular expression pattern to STRINGS is set as Match STRINGS, and STRINGS of order is set as front STRINGS and back STRINGS. (It does not operate on Windows.)

::sys <sub PATTERN STRINGS CORRESPDED-STRINGS AFTER-STRINGS>

The result of having transposed the portion which corresponded to STRINGS with the application of the regular expression pattern to Substitution STRINGS is set as Output STRINGS. Replacement is performed only once. (It does not operate on Windows.)

::sys <gsub PATTERN STRINGS CORRESPDED-STRINGS AFTER-STRINGS>

The result of having transposed the portion which corresponded to STRINGS with the application of the regular expression pattern to Substitution STRINGS is set as Output STRINGS. Replacement is performed into all the applicable portions of STRINGS. (It does not operate on Windows.)

::sys <split VAR STRINGS [DELIMITER]>

What divided STRINGS by the DELIMITER and was set to LIST is set as VAR. When the delimiter is not specified, it is divided with a blank and a tab.

::sys <toupper VAR STRINGS>

A character string is made into a capital letter.

::sys <tolower VAR STRINGS>

A character string is made into a small letter.

::sys <length VAR LIST>

The length of LIST is set as VAR.

::sys <random VAR>

A random number is assigned to VAR.

::sys <sin VAR RADIAN>

::sys <cos VAR RADIAN>

::sys <tan VAR RADIAN>

Trigonometric functions

::sys <asin VAR VAL>

::sys <acos VAR VAL>

::sys <atan VAR VAL>

Inverse trigonometric function

::sys <sinh VAR RADIAN>

::sys <cosh VAR RADIAN>

::sys <tanh VAR RADIAN>

Hyperbolic trigonometric functions

::sys <asinh VAR VAL>

::sys <acosh VAR VAL>

::sys <atanh VAR VAL>

Hyperbolic inverse trigonometric function

::sys <log VAR VAL>

::sys <log10 VAR VAL>

::sys <exp VAR VAL>

::sys <pow VAR VAL1 VAL2>

Logarithmic function

::sys <sqrt VAR VAL>

Square root

::sys <abs VAR VAL>

Absolute value

::sys <int VAR VAL>

Integral value

::sys <floor VAR VAL>

largest integral value not greater than argument

::sys <car VAR LIST>

::sys <cdr VAR LIST>

car, cdr of LIST

::sys <cons VAR LIST1 LIST2>

Connection of LIST

::sys <code CODE-NAME>

A setup of a character code.
UTF8, EUCJP, and SJIS can be specified.

::sys <char VAR STRINGS>

STRINGS is decomposed for every be character, and it is made LIST, and is set as VAR. A multibyte character like a Japanese character is also right, and it is every character.

::sys <byte VAR STRINGS>

STRINGS is decomposed for every character code, and it is made LIST, and is set as VAR.

::sys <asciichar VAR STRINGS>

::sys <utf8char VAR STRINGS>

::sys <eucchar VAR STRINGS>

::sys <sjischar VAR STRINGS>

STRINGS is decomposed for every character code, and it is made LIST, and is set as VAR.

::sys <concat VAR LIST>

Strings LIST is made to unite, STRINGS is compounded and it is set as VAR.

::sys <concatcode VAR LIST>

Character code LIST is made to unite, STRINGS is compounded and it is set as VAR.

::sys <bitand VAR NUMBER1 NUMBER2>

::sys <bitor VAR NUMBER1 NUMBER2>

::sys <bitxor VAR NUMBER1 NUMBER2>

::sys <bitnot VAR NUMBER1>

bit operation

```
::sys <shiftr VAR NUMBER SHIFT>
::sys <shiftr VAR NUMBER SHIFT>
```

The bit shift of an integral value. shiftr is shifted to the left and shiftr is shifted to the right.

```
::sys <eq ARG1 ARG2>
::sys <noteq ARG1 ARG2>
::sys <is ARG1 ARG2>
```

Comparison of ARG1 and ARG2

```
::sys <getc VAR>
```

One character is inputted into VAR.

```
::sys <putc CHAR>
```

One character is outputted.

```
::sys <getline VAR [PRED...]>
```

One line is inputted and it is set as VAR. When PRED is set up, PRED is performed by considering a tmp file as an input.

```
::sys <syntax STRINGS PRED...>
```

A predicate is performed by making a character string into an input file.

```
::sys <tmpfile VAR>
```

Temporary FILE NAME is set as VAR

```
::sys <openr FILE-NAME PRED...>
```

The file of FILE-NAME is opened in postscript reading, and PRED is performed.

```
::sys <openw FILE-NAME PRED...>
```

The file of FILE-NAME is opened in postscript writing, and PRED is performed.

```
::sys <openwp FILE-NAME PRED...>
```

The file of FILE-NAME is opened in postscript writing, and PRED is performed.

```
::sys <gettime VAR>
```

The present time is set as a variable by a micro second bit.

```
::sys <time VAR>
```

List of (the user time, sys time, and elapsed time)
of the predicate under execution is set as a variable.

::sys <date>

Time is set as a variable.

::sys <countnode VAR>

The number of nodes currently used is set as VAR.

::sys <gc>

A garbage collector is started.