Sanscript™

Fully Visual Scripting

Environment

Function Reference Guide

This guide provides information on using the functions provided with the Sanscript Fully Visual Scripting Environment Version 2 for Windows 95, Windows 98, and Windows NT 4.0 or higher.

January 1999

Northwoods Software Corporation
142 Main St.
Nashua, NH 03060

http://www.nwoods.com

January 1999

CONTENTS

PREFACE

Purpose of this guide

This guide describes the functions available in the Catalog of the Sanscript visual scripting tool.

For further information about using the Sanscript tool, see the Sanscript User Guide.

Who should use this guide

This guide is intended for a broad audience, including system administrators, application programmers, and any other user of the tool.

Structure of this guide

This guide is organized as follows:

- Introduction describes the format used to describe the Sanscript functions.
- Examples and Templates Cabinet describes the functions provided to assist understanding and programming construction.
- General File Cabinet describes the functions provided for general programming use.

Assumptions

This manual assumes you are familiar with Windows and Windows NT concepts and terminology. If you are not, please refer to your Windows documentation or online help.

Conventions

This manual uses the following conventions.

General conventions

The list below shows special kinds of formatting used in this manual.

Note:	Clarification or exception.
Caution:	Warning about conditions that could cause unexpected damage to data, software, or equipment.
`Courier`	Text you enter from the keyboard or view on the screen
Arial bold	Command
Key+Key	Key combination. Hold down the first key and press the second key.

Mouse conventions

The list below defines key words associated with mouse operations and movements in this manual.

Point	Position the mouse cursor so that the tip rests directly on the screen object.
Click	Press and release the left mouse button.
Double-click	Quickly press and release twice the left mouse button.
Right-click	Press and release the right mouse button.
Drag	Point, press and hold the left mouse button, move the cursor to the new location, release the left mouse button.

1. INTRODUCTION

This manual describes all the functions supplied in the Sanscript Catalog. Use this manual to learn what supplied functions are available and what calculations they perform.

The description for each file cabinet lists the folders that file cabinet contains.

The description for each folder lists the functions that folder contains.

The description for each function includes the following information:

- The name of the function
- A description of what the function does
- The names and datatypes of each inlet (input) and outlet (output)
- The name of the folder that contains the function

2. EXAMPLES AND TEMPLATES FILE CABINET

The Examples and Templates file cabinet contains functions that illustrate how to use Sanscript. You can create your own functions by copying these functions to the User file cabinet and changing them as you wish.

Contains folders:

- Simple Examples folder
- System Examples folder
- Example Functions folder
- Math Quiz folder
- Batch Examples folder
- Templates folder

Simple Examples Folder

Simple programs that demonstrate basic features.

Contents of folder:

- Hello World
- Do you want to say hello?
- Fibonacci
- Compound Interest
- Test Progress Bar

Folder: Examples and Templates

Hello World

A very simple program that demonstrates how to display a message.

Inlets: none.

Outlets: none.

Folder: Simple Examples

Do you want to say hello?

A simple program that demonstrates a Pick One.

Inlets: none.

Outlets: none.

Folder: Simple Examples

Fibonacci

A program that computes the Fibonacci sequence: 1, 2, 3, 5, 8, 13...
It demonstrates the use of a Repeat.

Inlets: none.

Outlets: none.

Folder: Simple Examples

Compound Interest

Demonstrate Repeats and Packages.

Inlets: none.

Outlets: none.

Folder: Simple Examples

Test Progress Bar

Demonstrate use of a progress bar and a status line.

Inlets: none.

Outlets: none.

Folder: Simple Examples

System Examples Folder

Simple programs that examine the hardware and operating system.

Contents of folder:

- Disk Space Used
- Directory Tree Listing
- Show Disk Drives
- Show System Info
- Show Memory Info
- Show Services
- Test Send Keys
- Show Registry File Types

Folder: Examples and Templates

Disk Space Used

Find out how much disk space is used by a set of files.

Inlets: none.

Outlets: none.

Folder: System Examples

Directory Tree Listing

Show a listing of all the files matching a given name.

Inlets: none.

Outlets: none.

Folder: System Examples

Show Disk Drives

Show a listing of the disk drives available on the system.

Inlets: none.

Outlets: none.

Folder: System Examples

Show System Info

Present information about this computer system.

Inlets: none.

Outlets: none.

Folder: System Examples

Show Memory Info

Present information about the amount of memory on this computer.

Inlets: none.

Outlets: none.

Folder: System Examples

Show Services

Show a list of the services on this NT system.

Inlets: none.

Outlets: none.

Folder: System Examples

Test Send Keys

Play an AVI file by controlling the Media Player using Send Keys.

Inlets: none.

Outlets: none.

Folder: System Examples

Show Registry File Types

Present a list of the file extensions under HKEY_CLASSES_ROOT.

Inlets: none.

Outlets: none.

Folder: System Examples

Example Functions Folder

Simple functions that might be useful.

Contents of folder:

- InRange?
- StringSubst
- Round to Decimal Fraction
- Fill with Blanks

Folder: Examples and Templates

InRange?

Determine if a decimal is within a given range. It will be out of the range if it is less than or equal to the Low number, or if it is greater than or equal to the High number.

Inlets:

Low (Decimal)

CurrentValue (Decimal)

High (Decimal)

Outlets:

InRange (Boolean)

OutRange (Boolean)

Folder: Example Functions

StringSubst

Function to substitute some text for another in a bigger text string. The input text must contain a question mark ("?"), which will be replaced with the SUBST text.

Inlets:

text (Text)

subst (Text)

Outlets:

text (Text)

Folder: Example Functions

Round to Decimal Fraction

Rounds decimal numbers to some fraction. By default rounds to 1/100, or two decimal places.

Inlets:

number (Decimal)

precision (Decimal) DEFAULT VALUE: 100

Outlets:

limited (Decimal)

Folder: Example Functions

Fill with Blanks

Produce a fixed length text. Given an input text string, this function pads with blanks (space characters) until it is SIZE characters long. If the input text is longer than SIZE, it is returned without truncation and without any appended blanks.

Inlets:

text (Text)

size (Integer) DEFAULT VALUE: 20

the minimum length of the padded text string; limit is 256

Outlets:

filled (Text)

input text plus blanks

Folder: Example Functions

Math Quiz Folder

A simple arithmetic drill program.

Contents of folder:

- Math Quiz
- Arith Test
- Generate Arith Num
- Ask Arith
- Check Arith
- Respond Arith

Folder: Examples and Templates

Math Quiz

Simple arithmetic drill

Inlets: none.

Outlets: none.

Folder: Math Quiz

Arith Test

Inlets:

Operation (Text) DEFAULT VALUE: +

Maximum (Integer) DEFAULT VALUE: 10

Outlets:

Continue (Boolean)

Folder: Math Quiz

Generate Arith Num

Inlets:

Minimum (Integer) DEFAULT VALUE: 0

Maximum (Integer) DEFAULT VALUE: 10

Outlets:

Random Number (Integer)

Folder: Math Quiz

Ask Arith

Inlets:

First number (Integer)

Operation (Text)

Second number (Integer)

Outlets:

Answer (Integer)

Folder: Math Quiz

Check Arith

Inlets:

First number (Integer)

Operation (Text)

Second number (Integer)

Response Number (Integer)

Outlets:

Correct (Boolean)

Correct number (Integer)

Folder: Math Quiz

Respond Arith

Inlets:

Correct (Boolean)

Correct number (Integer)

Outlets:

Continue (Boolean)

Folder: Math Quiz

Batch Examples Folder

Example programs that run without user interaction.
Use the File | Make Application command to produce a .BAT file
that may be called from a DOS command line or as a batch process.

Contents of folder:

- Report File Space Used
- Report Full Disks
- Generate Report Pathname
- Change Sounds in Registry

Folder: Examples and Templates

Report File Space Used

Produce a report listing all the files with a given extension, their
sizes, modified dates, and creation dates. The report is placed
in the given directory, under a subdirectory named FileSpaceUsed,
with a name that is the name of the computer.

The default file extension to search for is TMP.

If a report directory is not supplied, the current working directory
will be used to hold the FileSpaceUsed subdirectory.

Inlets:

report directory (Pathname) DEFAULT VALUE:

Network path for a directory where the FileSpaceUsed subdirectory will hold the report; the filename will be <machine-name> .LOG; defaults to the current working directory on this machine

extension (Text) DEFAULT VALUE: TMP

file type to search for on all disk drives

Outlets: none.

Folder: Batch Examples

Report Full Disks

Produce a report listing all disks that have less than some
percentage space free. The report is placed in the given directory,
under a subdirectory named FullDisks, with a name that is the
name of the computer.

The default free percentage is 5%.

If a report directory is not supplied, the current working directory
will be used to hold the FullDisks subdirectory.

Inlets:

report directory (Pathname) DEFAULT VALUE:

network path for the FullDisks subdirectory that will hold the report; filename will be <machine-name> .LOG; defaults to the current working directory on this machine

percent free (Integer) DEFAULT VALUE: 5

if the percent free on the disk is less than this value, add it to the report

Outlets: none.

Folder: Batch Examples

Generate Report Pathname

Get the pathname of the report file, given:
a directory path, typically a network share directory
(this defaults to the current directory of the current machine)
a subdirectory name, typically the name of the report
generating program
The file name will be the name of the machine with a LOG
file extension.

Inlets:

sub dir name (Text)

Name of subdirectory for report

dir path (Pathname)

Parent directory for subdirectory to hold reports

Outlets:

log file (Pathname)

pathname for report log file

Folder: Batch Examples

Change Sounds in Registry

A sample program for changing the registry.
The process must have permission for making the changes.

Inlets: none.

Outlets: none.

Folder: Batch Examples

Templates Folder

Collections of functions organized to do certain common tasks.
Just drop one into your flowgram and customize for your needs.
Look for Todo comments inside Repeats or PickOnes.

Contents of folder:

- Read File Template
- Modify File Template
- Write File Template
- Build List Template
- Filter List Template

Folder: Examples and Templates

Read File Template

A template for reading the lines of a file.
Drop this into your flowgram and complete the body of the Repeat.

Inlets: none.

Outlets: none.

Folder: Templates

Modify File Template

A template for copying a file while changing some or all of its lines.
Drop this into your flowgram and complete the body of the Repeat.

Inlets: none.

Outlets: none.

Folder: Templates

Write File Template

A template for creating a file line-by-line.
Drop this into your flowgram and complete the body of the Repeat.

Inlets: none.

Outlets: none.

Folder: Templates

Build List Template

A template for creating a list an item at a time.
Drop this into your flowgram and complete the body of the Repeat.

Inlets: none.

Outlets: none.

Folder: Templates

Filter List Template

A template for creating a list by selecting items from an existing list.
Drop this into your flowgram and complete the body of the Repeat.

Inlets: none.

Outlets: none.

Folder: Templates

3. General file cabinet

The General file cabinet contains generally useful functions for manipulating text and numbers and lists and other data types, and functions for dealing with the system environment and the user interface. You can use these functions to create your own more complex functions.

Contains folders:

- User Interface folder
- Text folder
- Integer folder
- Decimal folder
- List folder
- File I/O folder
- Directory folder
- Logical folder
- System folder
- Time folder
- Registry folder
- Error folder
- DDE folder
- Exp & Trig folder
- Conversions folder
- Language folder

User Interface folder

Functions that allow your program to display windows for interacting with the user.

Contents of folder:

- Display Message
- Ask Yes/No
- Prompt For Text
- File Open Box
- File Save Box
- Set Integer Progress
- Set Decimal Progress
- Set Status Line
- Set Monitor Visibility
- Disable Close App

Folder: General

Display Message

Display the Message in a small window; wait for user to dismiss window.

Inlets:

msg (Text)

Outlets: none.

Folder: User Interface

Ask Yes/No

Display Prompt in a window; return user's response, Yes (TRUE) or No (FALSE).

Inlets:

prompt (Text)

Outlets:

Yes/No (Boolean)

Folder: User Interface

Prompt For Text

Display Prompt in a window; return what user types.

Inlets:

prompt (Text)

Outlets:

text (Text)

Folder: User Interface

File Open Box

Put up a File Open Box and return user response.
Allows the user to select a file by searching a directory.
The user may specify that the file is to be opened Read-Only.
The user is also given the option of creating the file if it doesn't exist.

Inlets:

default filename (Pathname) DEFAULT VALUE:

Default filename to display to user

default extension (Text) DEFAULT VALUE:

Default extension if user enters none, for example: txt

file types (Text) DEFAULT VALUE:

File types to search for, for example: *.txt

Outlets:

path (Pathname)

user selected full pathname for file

OK? (Boolean)

TRUE if the user pressed the OK button

status (Error)

Status information

read only? (Boolean)

True if the user checked Read-Only

Folder: User Interface

File Save Box

Put up a File Save As box and return user's choice of filename.

Allows the user to view a directory contents and choose a name and type for a file.
The filename ultimately chosen by the user is returned.

Inlets:

default filename (Pathname) DEFAULT VALUE:

Default filename to display to user

default extension (Text) DEFAULT VALUE:

Default extension if user enters none, for example: txt

file types (Text) DEFAULT VALUE:

File types to search for, for example: *.txt

Outlets:

path (Pathname)

user selected full pathname for file

OK? (Boolean)

TRUE if the user pressed the OK button

status (Error)

Status information

Folder: User Interface

Set Integer Progress

Change the progress bar in the application's monitor window.

Inlets:

part (Integer)

total (Integer)

Outlets: none.

Folder: User Interface

Set Decimal Progress

Change the progress bar in the application's monitor window.

Inlets:

part (Decimal)

total (Decimal)

Outlets: none.

Folder: User Interface

Set Status Line

Display a line of text along with the progress bar.

Inlets:

line (Text)

Outlets: none.

Folder: User Interface

Set Monitor Visibility

Hide or show the monitor, containing the status line and progress bar.

Inlets:

vis? (Boolean) DEFAULT VALUE: TRUE

Outlets: none.

Folder: User Interface

Disable Close App

Normally the application's monitoring window, containing a status line and progress bar, may be closed by the user at any time, terminating that application. This function allows the program to delay exiting the application while the program is doing a number of operations that must not be interrupted by the user closing the application.

Inlets:

protect (Boolean) DEFAULT VALUE: TRUE

Outlets:

prev (Boolean)

Folder: User Interface

Text folder

Functions for manipulating text strings.

Contents of folder:

- Concatenate Text
- Length of Text
- Append Text
- Compare Text Exact
- Compare Text NoCase
- Upcase Text
- Downcase Text
- Capitalize Text
- Split Text
- Search & Split Text
- Trim Text
- Substitute Text
- Format Integers
- Format Decimals
- Write List As Text Alias
- Read List Flexibly Alias
- Read List Exactly Alias
- Write Record As Text
- Subtract Text
- Search Text Reverse
- Search Text For Many
- Char

Folder: General

Concatenate Text

Return text abc by merging text a, b, and c.

Inlets:

a (Text) DEFAULT VALUE:

b (Text) DEFAULT VALUE:

c (Text) DEFAULT VALUE:

Outlets:

abc (Text)

Folder: Text

Length of Text

Return the number of characters in the given text.

Example: length of "abc" is 3.

Inlets:

s (Text)

Outlets:

len (Integer)

Folder: Text

Append Text

Make a text string consisting of all the inputs put together in order.
There is always an unconnected inlet so one can append more text.

Example: "a", "ab", "abc" produces "aababc".

Inlets:

(List of Text) VARYING NUMBER OF INLETS DEFAULT VALUE:

Outlets:

(Text)

Folder: Text

Compare Text Exact

Compare text (case matters).
a==b is TRUE if its inputs are equal.
a<b is TRUE if A comes alphabetically before B.
a>b is TRUE if A comes alphabetically after B.

Inlets:

a (Text)

b (Text)

Outlets:

a==b (Boolean)

a<b (Boolean)

a>b (Boolean)

Folder: Text

Compare Text NoCase

Compare text, ignoring case.
a==b is TRUE if inputs are equal.
a<b is TRUE if A comes alphabetically before B.
a>b is TRUE if A comes alphabetically after B.

Inlets:

a (Text)

b (Text)

Outlets:

a==b (Boolean)

a<b (Boolean)

a>b (Boolean)

Folder: Text

Upcase Text

Convert text to all upper case.

Example: "aB23cD" produces "AB23CD".

Inlets:

a (Text)

Outlets:

A (Text)

Folder: Text

Downcase Text

Convert text to all lower case.

Example: "aB23cD" produces "ab23cd".

Inlets:

A (Text)

Outlets:

a (Text)

Folder: Text

Capitalize Text

Capitalize all the words in some text by upcasing the first letter and
any letters after a whitespace, and downcasing all other letters.

Example: "a GREAT paradigm" ==> "A Great Paradigm"

Inlets:

teXt (Text)

Outlets:

Text (Text)

Folder: Text

Split Text

Split text into 3 parts: Before, Middle, and After.
The text is numbered starting with 1.
Middle Start is the starting position of the Middle.
Middle Start+Length-1 is the last position of the Middle.
A Middle Start of 0 or less is the same as 1.
After extracting the Middle portion, Before and After are set
to what text is on either side, if any.

Example: "abcdef", middle start = 2, length = 2
before = "a", middle = "bc", after = "def"

Inlets:

text (Text)

middle start (Integer)

length (Integer)

Outlets:

before (Text)

middle (Text)

after (Text)

Folder: Text

Search & Split Text

Search and split text into: Before, Middle, and After.
The Body input is scanned to locate the Search text.
If found, Middle is set to the Search text and Before and After are set to the text on either side, if any.
If the Search text is not found, Before is set to the Body, and Middle and After are null.

Example: body = "abcabcabc", search = "bc"
before = "a", middle = "bc", after = "abcabc"

Inlets:

body (Text)

search (Text)

Outlets:

found (Boolean)

before (Text)

middle (Text)

after (Text)

Folder: Text

Trim Text

Remove specified subtext from the start and end of text.
Chars to Trim specifies the subtext to be removed from the begining and/or end of the text.
If Leading is true, all matching subtext at the start of the text is removed.
If Trailing is true, all matching subtext at the end of Text is removed.
Any matching subtext within the text is never affected.

Example: text = "   lots of spaces   ", charstotrim = " ",
leading = TRUE, trailing = TRUE, then result = "lots of spaces"

Inlets:

string (Text)

chars to trim (Text) DEFAULT VALUE:

if no characters to trim are provided, whitespace will be trimmed

leading (Boolean) DEFAULT VALUE: TRUE

trailing (Boolean) DEFAULT VALUE: TRUE

Outlets:

trimmed (Text)

Folder: Text

Substitute Text

Search for subtext in text and replace it with new subtext.
Old substr is the old subtext being looked for.
New Substr is the subtext that is to replace it.
If the Old Substr is not found in String, then the output is the same as String.

Example: text = "aabbaaaa", new substr = "a", old substr = "aa"
result = "abbaa"

Inlets:

text (Text)

new substr (Text)

old substr (Text)

Outlets:

text (Text)

Folder: Text

Format Integers

Insert integers into text, replacing up to 3 special markers
with the textual representation of the integer values.
The markers may be %ld for decimal integers, %lx for hexadecimal integers, or %lo for octal integers.

Inlets:

format %ld (Text)

i1 (Integer) DEFAULT VALUE: 0

i2 (Integer) DEFAULT VALUE: 0

i3 (Integer) DEFAULT VALUE: 0

Outlets:

text (Text)

Folder: Text

Format Decimals

Insert decimals into text, replacing up to 3 special markers
with the textual representation of the decimal values.
The special markers may be %e for exponential format numbers,
%f for non-exponential format numbers, or %g for whichever is
more compact depending on the value.
You may specify the number of digits after the decimal point by
using a marker like %.2f, which indicates two digits after the
decimal point.

Inlets:

format %g (Text)

f1 (Decimal) DEFAULT VALUE: 0

f2 (Decimal) DEFAULT VALUE: 0

f3 (Decimal) DEFAULT VALUE: 0

Outlets:

text (Text)

Folder: Text

Write Record As Text

A simple function for writing the contents of any record as text.

Inlets:

rec (Generic Record)

Outlets:

text (Text)

Folder: Text

Subtract Text

Remove a prefix and/or suffix from some text, if present.
Examples: text="logfile.log", subtext="log"
If leading=TRUE, trailing=TRUE, then result="file."
If leading=FALSE, trailing=TRUE, then result="logfile."
If leading=TRUE, trailing=FALSE, then result="file.log"

Inlets:

text (Text)

subtext (Text)

leading (Boolean) DEFAULT VALUE: TRUE

trailing (Boolean) DEFAULT VALUE: TRUE

Outlets:

res (Text)

Folder: Text

Search Text Reverse

This is just like Search & Split Text, but searches backwards from the end of the text.
Example: body = "d:\some\dir\path", search = "\"
before = "d:\some\dir", middle = "\", after = "path"

Inlets:

body (Text)

search (Text)

Outlets:

found (Boolean)

before (Text)

middle (Text)

after (Text)

Folder: Text

Search Text For Many

This is like Search & Split Text, but instead of looking for a single search string, it looks for one of many, as provided in a list of text. This returns the shortest resulting text before a found search text; i.e. the text in the list found earliest in the body.

Example: body = "aabbccaabbcc", list = ("cc","bb")
before = "aa", middle = "bb", after = "ccaabbcc"

Inlets:

body (Text)

list (List of Text)

Outlets:

found (Boolean)

before (Text)

middle (Text)

after (Text)

Folder: Text

Char

Produce text containing a single character with the given character code. Useful for producing text strings containing characters such as NewLine or Tab or other control characters needed in some files.

Inlets:

int (Integer)

Outlets:

int (Text)

Folder: Text

Integer folder

Functions for doing integer arithmetic and comparisons.

Contents of folder:

- Plus
- Minus
- Times
- Quotient
- Remainder
- Maximum
- Minimum
- Increment
- Decrement
- Negate
- Absolute Value
- LessThan
- GreaterThan
- LessThan Or Equal
- GreaterThan Or Equal
- Equal
- NotEqual
- Random Number
- Bitwise AND
- Bitwise OR
- Bitwise XOR
- Bitwise Complement

Folder: General

Plus

Return the sum of two integers.

Inlets:

i (Integer)

j (Integer)

Outlets:

i+j (Integer)

Folder: Integer

Minus

Return the difference of two integers by subtracting J from I.

Inlets:

i (Integer)

j (Integer)

Outlets:

i-j (Integer)

Folder: Integer

Times

Return the product of two integers by multiplying I and J.

Inlets:

i (Integer)

j (Integer)

Outlets:

*i*j* (Integer)

Folder: Integer

Quotient

Return the quotient of two integers by dividing I by J.
The result is the whole number of times J goes into I.
It is an error if J is zero.

Inlets:

i (Integer)

j (Integer)

Outlets:

i/j (Integer)

Folder: Integer

Remainder

Divide I by J and return the remainder.

Inlets:

i (Integer)

j (Integer)

Outlets:

i%j (Integer)

Folder: Integer

Maximum

Return the number with the greatest value.

Inlets:

i (Integer) DEFAULT VALUE: 0

j (Integer) DEFAULT VALUE: 0

Outlets:

i max j (Integer)

Folder: Integer

Minimum

Return the number with the smallest value.

Inlets:

i (Integer) DEFAULT VALUE: 0

j (Integer) DEFAULT VALUE: 0

Outlets:

i min j (Integer)

Folder: Integer

Increment

Return a value one larger than the given number.

Inlets:

i (Integer)

Outlets:

i+1 (Integer)

Folder: Integer

Decrement

Return a number one less than the given number.

Inlets:

i (Integer)

Outlets:

i-1 (Integer)

Folder: Integer

Negate

If the number is zero, return zero. Otherwise return the same number but of opposite sign.

Inlets:

i (Integer)

Outlets:

-i (Integer)

Folder: Integer

Absolute Value

If the number is zero or positive, return it. Otherwise return the negation of the number (which will be positive).

Inlets:

i (Integer)

Outlets:

|i| (Integer)

Folder: Integer

LessThan

Return TRUE if the value of I is less than J.

Inlets:

i (Integer)

j (Integer)

Outlets:

i<j (Boolean)

Folder: Integer

GreaterThan

Return TRUE if the value of I is more than J.

Inlets:

i (Integer)

j (Integer)

Outlets:

i>j (Boolean)

Folder: Integer

LessThan Or Equal

Return TRUE if the value of I is less than or the same as J.

Inlets:

i (Integer)

j (Integer)

Outlets:

i<=j (Boolean)

Folder: Integer

GreaterThan Or Equal

Return TRUE if the value of I is more than or the same as J.

Inlets:

i (Integer)

j (Integer)

Outlets:

i>=j (Boolean)

Folder: Integer

Equal

Return TRUE if the values of I and J are the same, FALSE otherwise.

Inlets:

i (Integer)

j (Integer)

Outlets:

i==j (Boolean)

Folder: Integer

NotEqual

Return TRUE if the values of I and J are different, FALSE otherwise.

Inlets:

i (Integer)

j (Integer)

Outlets:

i!=j (Boolean)

Folder: Integer

Random Number

Return a random integer in the range 1..Max.

Inlets:

max (Integer) DEFAULT VALUE: 100

the upper limit for the resulting random number

Outlets:

?i (Integer)

a not-very-predictable integer with a value between 1 and MAX (inclusive)

Folder: Integer

Bitwise AND

Return the bitwise logical And of the two integers.

Example: 10 AND 12 ==> 8 (binary 1010 AND 1100 ==> 1000)

Inlets:

i (<u>Integer</u>)

j (<u>Integer</u>)

Outlets:

i&j (<u>Integer</u>)

Folder: Integer

Bitwise OR

Return the bitwise logical inclusive OR of the two integers.

Example: 10 OR 12 ==> 14 (binary 1010 OR 1100 ==> 1110)

Inlets:

i (<u>Integer</u>)

j (<u>Integer</u>)

Outlets:

i|j (<u>Integer</u>)

Folder: Integer

Bitwise XOR

Return the bitwise logical exclusive OR of the two integers.

Example: 10 XOR 12 ==> 6 (binary 1010 XOR 1100 ==> 0110)

Inlets:

i (<u>Integer</u>)

j (<u>Integer</u>)

Outlets:

i^j (<u>Integer</u>)

Folder: Integer

Bitwise Complement

Returns the bitwise complement of the integer.

Example: COMPL 2 ==> -3 (binary 00...010 ==> 11...101)

Inlets:

i (Integer)

Outlets:

~*i* (Integer)

Folder: Integer

Decimal folder

Functions for doing decimal (floating point) arithmetic and comparisons.

Contents of folder:

- Plus Decimal
- Minus Decimal
- Times Decimal
- Divide Decimal
- Maximum Decimal
- Minimum Decimal
- Absolute Value Decimal
- Negate Decimal
- Power Decimal
- Reciprocate Decimal
- Square Root Decimal
- LessThan Decimal
- GreaterThan Decimal
- LessThan Or Equal Decimal
- GreaterThan Or Equal Decimal
- Equal Decimal
- NotEqual Decimal
- Decimal To Integer Alias
- Round Decimal To Integer Alias

Folder: General

Plus Decimal

Return the sum of the inputs.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f+g (Decimal)

Folder: Decimal

Minus Decimal

Return the difference by subtracting G from F.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f-g (Decimal)

Folder: Decimal

Times Decimal

Multiply the inputs.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

*f*g* (Decimal)

Folder: Decimal

Divide Decimal

Return the quotient by dividing F by G.
If G is zero, the result will not be an error but INFINITY,
unless F is also zero, in which case the result is INDETERMINATE.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f/g (Decimal)

Folder: Decimal

Maximum Decimal

Return the larger value.

Inlets:

f (Decimal) DEFAULT VALUE: 0

g (Decimal) DEFAULT VALUE: 0

Outlets:

f max g (Decimal)

Folder: Decimal

Minimum Decimal

Return the smaller value.

Inlets:

f (Decimal) DEFAULT VALUE: 0

g (Decimal) DEFAULT VALUE: 0

Outlets:

f min g (Decimal)

Folder: Decimal

Absolute Value Decimal

Return the number if it is zero or positive, otherwise return
the negation of the negative value (a positive number).

Inlets:

f (Decimal)

Outlets:

|f| (Decimal)

Folder: Decimal

Negate Decimal

If the number is zero, return zero.
Otherwise return the same number but with the opposite sign.

Inlets:

f (Decimal)

Outlets:

-f (Decimal)

Folder: Decimal

Power Decimal

Return the result of an exponentiation: F raised to the power of G.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f^g (Decimal)

Folder: Decimal

Reciprocate Decimal

Return 1 divided by the number. If the number is zero, the result is INFINITY.

Inlets:

f (Decimal)

Outlets:

1/f (Decimal)

Folder: Decimal

Square Root Decimal

Return the square root of the number.
If the number is negative, the result is INDETERMINATE.

Inlets:

f (Decimal)

Outlets:

sqrt(f) (Decimal)

Folder: Decimal

LessThan Decimal

Return TRUE if F has a value less than G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f<g (Boolean)

Folder: Decimal

GreaterThan Decimal

Return TRUE if F has a value greater than G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f>g (Boolean)

Folder: Decimal

LessThan Or Equal Decimal

Return TRUE if F has a value less than or equal to G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f<=g (Boolean)

Folder: Decimal

GreaterThan Or Equal Decimal

Return TRUE if F has a value greater than or equal to G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f>=g (Boolean)

Folder: Decimal

Equal Decimal

Return TRUE if F has the same value as G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f==g (Boolean)

Folder: Decimal

NotEqual Decimal

Return TRUE if F has a value different from G, FALSE otherwise.

Inlets:

f (Decimal)

g (Decimal)

Outlets:

f!=g (Boolean)

Folder: Decimal

List folder

Functions for manipulating ordered lists of values.

Contents of folder:

- List Of
- Make List of Items
- Length of List
- List Item
- Write List As Text
- Read List Exactly
- Read List Flexibly
- Add Last Item
- Remove Last Item
- Last Item
- Member In List
- Insert In List
- Remove At In List
- Compare List
- Replace List Item
- SubList
- Join Lists
- Reverse List
- Sort List
- Sort List of Records
- Remove From List
- Substitute In List
- Union Of Lists
- Intersection Of Lists
- Difference Of Lists
- Remove List Duplicates

Folder: General

List Of

Create a list of a particular data type and specified length,
filled with a given value.

Inlets:

init (***any***)

The Item Type of the Type of *list*

Initial value for each item in the list to be created

len (Integer)

Length of the list to be created

Outlets:

list (Generic List)

A List Type whose Item Type is the Type of *init*

Folder: List

Make List of Items

Create a list from all the items linked to the inlets.
The first item linked determines the data type of all items of the list.
Each time an item is linked in, an additional inlet is created for
additional items.

Inlets:

(Generic List) VARYING NUMBER OF INLETS

The same Type as *list*

An individual list item

Outlets:

list (Generic List)

The same Type as

Folder: List

Length of List

Return the length of the list.

Inlets:

list (Generic List)

Outlets:

len (Integer)

The number of items in the list; the length is also the position of the last item in the list

Folder: List

List Item

Return the item at position POS in the List.
The position starts at 1 for the first item.

Inlets:

list (Generic List)

A List Type whose Item Type is the Type of *item*

pos (Integer)

The position in the list of the item to retrieve. The first item is at position 1.

Outlets:

item (***any***)

The Item Type of the Type of *list*

The value of the item at position POS in the list.

Folder: List

Write List As Text

Convert a list to text.
Output s starts with Prefix, then each list item is converted to text
and followed by the Separator text input.
Finally, the Suffix is appended.

Inlets:

list (Generic List)

prefix (Text) DEFAULT VALUE:

a text to insert before any of the list items

sep (Text) DEFAULT VALUE: ,

the text to insert between items

suffix (Text) DEFAULT VALUE:

a text to insert after all of the list items

Outlets:

s (Text)

all the items of the list written as text

Folder: List

Read List Exactly

Convert properly formatted text to a list.
First the specified Prefix and Suffix are removed from the String.
Then, the remaining text is split into parts by finding the exact Separator text string.
The remaining pieces are then converted into items in a List.
This function is the opposite of Write List As Text.

Inlets:

text (Text)

this contains the list to be read, represented as text

prefix (Text) DEFAULT VALUE:

start reading items of the list after this prefix

sep text (Text) DEFAULT VALUE: ,

each occurrence of this exact text string is treated as a separator between one list item and the next

suffix (Text) DEFAULT VALUE:

an occurrence of this text indicates the end of the list

Outlets:

list (List of Text)

Folder: List

Read List Flexibly

Convert properly formatted text to a list.
First the specified Prefix and Suffix are removed from the String.
Then, the remaining text is split into parts by finding any of the characters specified by the Separator text. All consecutive sequences of any of those characters are removed.
The remaining pieces are then converted into the List.
This function is the opposite of Write List As Text, except that the Separator text is treated differently.

Inlets:

text (Text)

This contains the list represented as a text string.

prefix (Text) DEFAULT VALUE:

start reading items of the list after this prefix

sep chars (Text) DEFAULT VALUE: ,

any number of any of the characters in this text string indicate the end of one item and the start of another

suffix (Text) DEFAULT VALUE:

stop reading items of the list when this suffix appears

Outlets:

list (List of Text)

Folder: List

Add Last Item

Add a new item to the end of a List.

Inlets:

list (Generic List)

The same Type as *list*

A List Type whose Item Type is the Type of *newend*

newend (***any***)

The Item Type of the Type of *list*

the value to add at the end of the list

Outlets:

list (Generic List)

The same Type as *list*

Folder: List

Remove Last Item

Remove the last item from a list, and return both the shortened list and what had been the last item of the list.

Inlets:

list (Generic List)

The same Type as *newlist*

A List Type whose Item Type is the Type of *oldlast*

Outlets:

oldlast (***any***)

The Item Type of the Type of *list*

what used to be the last item of the list

newlist (Generic List)

The same Type as *list*

the list without the last item

Folder: List

Last Item

Return the last item from a List.

Inlets:

list (Generic List)

A List Type whose Item Type is the Type of *last*

Outlets:

last (***any***)

The Item Type of the Type of *list*

Folder: List

Member In List

Search a list to see if value Val is present;
return its position in the list, or -1 if not present.

Inlets:

list (Generic List)

A List Type whose Item Type is the Type of *val*

val (***any***)

The Item Type of the Type of *list*

The value to look for in the list.

Outlets:

pos (Integer)

The first position of the list holding VAL, or -1 if VAL is not in the list.

Folder: List

Insert In List

Insert a new item in a list at position POS.
If POS is greater than the length of the list,
the item is added at the end of the list.

Inlets:

list (Generic List)

The same Type as *list*

A List Type whose Item Type is the Type of *val*

pos (Integer)

The 1-based position for where to insert the new value.

val (***any***)

The Item Type of the Type of *list*

Outlets:

list (Generic List)

The same Type as *list*

A list with all the same items as the input list, but one longer, containing VAL at POS.

Folder: List

Remove At In List

Delete the item at position POS from a List.
Items in the list are numbered starting with 1.

Inlets:

list (Generic List)

The same Type as *list*

pos (Integer)

The 1-based position of the item in the input list to remove from the list.

Outlets:

list (Generic List)

The same Type as *list*

A list with the same items as the input list, but one shorter, missing the item that had been at position POS.

Folder: List

Compare List

Compare two lists to see if they are equal to, less than, or greater than each other. The comparison is done item-by-item, in order.

Inlets:

a (Generic List)

The same Type as *b*

b (Generic List)

The same Type as *a*

Outlets:

a==b (Boolean)

TRUE if both lists have all the same items in the same order.

a<b (Boolean)

a>b (Boolean)

Folder: List

Replace List Item

Replace the item at position POS with a new value.

Inlets:

list (Generic List)

The same Type as *list*

A List Type whose Item Type is the Type of *val*

pos (Integer)

The position in the list to replace an item.

val (***any***)

The Item Type of the Type of *list*

The value to become the new item in the list.

Outlets:

list (Generic List)

The same Type as *list*

A list that is the same as the input list, but with item at position POS replaced with VAL.

Folder: List

SubList

Return the portion of a list starting at position Start and ending at End.

Inlets:

list (Generic List)

The same Type as *sub*

start (Integer) DEFAULT VALUE: 1

The position in the list to get the first item of the new list. (Positions start at 1.)

end (Integer) DEFAULT VALUE: 9999999

The last position of the input list to take items from for the new list.

Outlets:

sub (Generic List)

The same Type as *list*

A part of the input list.

Folder: List

Join Lists

Merge up to three lists.

Inlets:

prefix (Generic List)

The same Type as *middle*

The same Type as *suffix*

The same Type as *list*

The items in this list become the first items of the resulting list.

middle (Generic List)

The same Type as *prefix*

The items in this list go in the middle of the resulting list.

suffix (Generic List)

The same Type as *prefix*

The items in this list form the end of the resulting list.

Outlets:

list (Generic List)

The same Type as *prefix*

A list containing all the items of the input lists, in order.

Folder: List

Reverse List

Return a list with its items in reverse order.

Inlets:

list (Generic List)

The same Type as *rev*

Outlets:

rev (Generic List)

The same Type as *list*

A list with all the same items as the input, but in exactly opposite order.

Folder: List

Sort List

Sort a list. This is not useful for sorting lists of records--use
Sort List of Records.

Inlets:

list (Generic List)

The same Type as *sorted*

up? (Boolean) DEFAULT VALUE: TRUE

If TRUE, the items are sorted from smallest to largest.

Outlets:

sorted (Generic List)

The same Type as *list*

A list with all the same items as the input list, but with the items in a rearranged order.

Folder: List

Sort List of Records

Sort a list of records, based on the values of a particular field in those records. The field is specified by the one-based position of the field in the record.

Inlets:

list (Generic List)

The same Type as *sorted*

up? (Boolean) DEFAULT VALUE: TRUE

If TRUE, the items are sorted from smallest to largest.

fld# (Integer)

The position of the field in the record to compare with when sorting

Outlets:

sorted (Generic List)

The same Type as *list*

A list with all the same items as the input list, but with the items in a rearranged order.

Folder: List

Remove From List

Remove all items from the list that are equal to the given value.

Inlets:

list (Generic List)

The same Type as *list*

A List Type whose Item Type is the Type of *val*

val (***any***)

The Item Type of the Type of *list*

Outlets:

list (Generic List)

The same Type as *list*

A list that is the same as the input list, but perhaps shorter, with no items equal to VAL.

Folder: List

Substitute In List

If Oldval is present in the list, replace all such items with Newval.

Inlets:

list (Generic List)

The same Type as *list*

A List Type whose Item Type is the Type of *newval*

newval (***any***)

The same Type as *oldval*

The Item Type of the Type of *list*

The value that will take the place of OLDVAL in the resulting list.

oldval (***any***)

The same Type as *newval*

The value that will not appear in the resulting list, having been replaced by NEWVAL.

Outlets:

list (Generic List)

The same Type as *list*

A list that is the same as the input list, except for NEWVAL replacing OLDVAL.

Folder: List

Union Of Lists

Merge two lists, but eliminate duplicate values.

Inlets:

x (Generic List)

The same Type as *y*

The same Type as *union*

y (Generic List)

The same Type as *x*

Outlets:

union (Generic List)

The same Type as *x*

A list whose items can be found in either X or Y or both.

Folder: List

Intersection Of Lists

Return a list of all items that two lists have in common.

Inlets:

x (Generic List)

The same Type as *y*

The same Type as *common*

y (Generic List)

The same Type as *x*

Outlets:

common (Generic List)

The same Type as *x*

A list whose items are present in both X and Y.

Folder: List

Difference Of Lists

Return a list of items in X that are not also in Y -- subtract
off the items in common.

Inlets:

x (Generic List)

The same Type as *y*

The same Type as *x-y*

y (Generic List)

The same Type as *x*

Outlets:

x-y (Generic List)

The same Type as *x*

The same list as X, but without any of the items that are in Y.

Folder: List

Remove List Duplicates

Remove all but one item of same-valued items in a list.

Inlets:

list (Generic List)

The same Type as *set*

Outlets:

set (Generic List)

The same Type as *list*

a list without any duplicate values

Folder: List

File I/O folder

Functions for reading and writing lines of disk files.

Contents of folder:

- Open File For Read
- Open File For Write
- Open File For Append
- Close Input File
- Close Output File
- Write Line
- Read Line
- Read File As Text
- Write Text As File
- Restart Input File
- Restart Output File

Folder: General

Open File For Read

Open the named file for reading and returns an INPUT FILE object.
The INPUT FILE object can be used in other file reading operations.

Inlets:

filename (Pathname)

Outlets:

file (Input File)

status (Error)

status information

any? (Boolean)

TRUE if there are any characters in the file

Folder: File I/O

Open File For Write

Open the named file for writing and returns an OUTPUT FILE object.

The OUTPUT FILE object can be used in other file writing operations.

Inlets:

filename (Pathname)

Outlets:

file (Output File)

status (Error)

status information

Folder: File I/O

Open File For Append

Open the named file for appending and returns an OUTPUT FILE object.
The OUTPUT FILE object can be used in other file writing operations.

Inlets:

filename (Pathname)

Outlets:

file (Output File)

status (Error)

status information

Folder: File I/O

Close Input File

Close an input file.
The file must have been opened with Open File For Read.

Inlets:

file (Input File)

Outlets:

status (Error)

status information

Folder: File I/O

Close Output File

Close an output file.
The file must have been opened with Open File For Write or Open File for Append.

Inlets:

file (Output File)

Outlets:

status (Error)

status information

Folder: File I/O

Write Line

Write a line of text to a file.
The file must have been opened with Open File For Write, or Open File For Append.

Inlets:

file (Output File)

line (Text)

Outlets:

file (Output File)

Folder: File I/O

Read Line

Read a line from a file.
The file must have been opened with Open File for Read.

Inlets:

file (Input File)

Outlets:

file (Input File)

line (Text)

eof (Boolean)

Folder: File I/O

Read File As Text

Open a file and return its contents as output Text.

Inlets:

filename (Pathname)

Outlets:

string (Text)

status (Error)

status information

Folder: File I/O

Write Text As File

Create a file, or overwrite an existing one, and fill it with Text.

Inlets:

string (Text)

filename (Pathname)

Outlets:

status (Error)

status information

Folder: File I/O

Restart Input File

Reposition the reading point to the beginning of an input file.
The file must have been opened with Open File For Read.

Inlets:

file (Input File)

Outlets:

file (Input File)

Folder: File I/O

Restart Output File

Reposition the writing point to the beginning of an output file.
The file must have been opened with Open File For Write or Open File for Append.

Inlets:

file (Output File)

Outlets:

file (Output File)

Folder: File I/O

Directory folder

Functions for looking at disk directories and manipulating files.

Contents of folder:

- Text to Path
- Make Path
- Path Parts
- Get File Info
- Set File Info
- Exists Path
- List Directory
- Delete Path
- Copy Path
- Make New Directory
- Get Current Directory
- Set Current Directory
- Get Path Space Used
- Get Disk Space
- Delete File
- Rename File
- Temp Filename
- List All Drives
- Disk Type

Folder: General

Text to Path

Convert properly formatted Text to a Path.
Examples of properly formatted text:

```
c:\windows\system\*.*
*.*
*.bat
c:\
```

Inlets:

string (Text) DEFAULT VALUE:

Outlets:

path (Pathname)

Folder: Directory

This Function acts as an automatic conversion from Text to Pathname

Make Path

Make a valid pathname from text parts.
Example of valid text inputs:

Device:	C
Directory:	\WINDOWS\SYSTEM
Name:	MYAPP
Extension:	INI

Inlets:

device (Text) DEFAULT VALUE:

directory (Text) DEFAULT VALUE:

name (Text) DEFAULT VALUE:

extension (Text) DEFAULT VALUE:

Outlets:

path (Pathname)

Folder: Directory

Path Parts

Obtain the text parts of a pathname.
Example of outputs for some pathname:

Device:	C:
Directory:	\WINDOWS\SYSTEM
Name:	MYAPP
Extension:	INI

Inlets:

path (Pathname)

Outlets:

status (Error)

status information

device (Text)

directory (Text)

name (Text)

extension (Text)

Folder: Directory

Get File Info

Obtain the file attributes of a file or directory.
This function generates an error if the file or directory is not found.

Inlets:

path (Pathname)

Outlets:

status (Error)

status information

read (Boolean)

write (Boolean)

execute (Boolean)

hidden (Boolean)

archive (Boolean)

modified (Time)

accessed (Time)

created (Time)

dir? (Boolean)

Folder: Directory

Set File Info

Set the file attributes of a file or directory.
This function generates an error if the file or directory is not found.

Inlets:

path (Pathname)

read (Boolean) DEFAULT VALUE: TRUE

write (Boolean) DEFAULT VALUE: TRUE

execute (Boolean) DEFAULT VALUE: TRUE

hidden (Boolean) DEFAULT VALUE: FALSE

archive (Boolean) DEFAULT VALUE: FALSE

time modified (Time) DEFAULT VALUE: <unknown Time>

time accessed (Time) DEFAULT VALUE: <unknown Time>

Outlets:

status (Error)

status information

Folder: Directory

Exists Path

Determine if the file or directory specified by pathname actually exists.
Will search subdirectories if Subdir is true.

Inlets:

path (Pathname)

subdir (Boolean) DEFAULT VALUE: FALSE

Outlets:

exists (Boolean)

first path found (Pathname)

Folder: Directory

List Directory

Given a pathname, returns a list of all the files or directories found.
The pathname may be partial, such as C:\*.TXT.
The inputs Include files, Include dirs, and subdir determine if files and/or directories are to be listed, and/or whether to search all subdirectories below the pathname.

Inlets:

path (Pathname)

include files (Boolean) DEFAULT VALUE: TRUE

include dirs (Boolean) DEFAULT VALUE: FALSE

subdir (Boolean) DEFAULT VALUE: FALSE

Outlets:

pathnames (List of Pathnames)

status (Error)

status information

Folder: Directory

Delete Path

Delete ALL files and/or directories matching the specified pathname.
If Subdir is true, all subdirectories below pathname are searched for matches.

Inlets:

path (Pathname)

subdir (Boolean) DEFAULT VALUE: FALSE

Outlets:

status (Error)

status information

Folder: Directory

Copy Path

Copy Src File to Dst File.
If Append is TRUE, the resulting file consists of all of the original
Destination file followed by all of the Source file.
Currently this function does not accept wildcards.

Inlets:

src file (Pathname)

dst file (Pathname)

append (Boolean) DEFAULT VALUE: FALSE

Outlets:

status (Error)

status information

Folder: Directory

Make New Directory

Create a new directory.
If the directory already exists, and Delete Existing Files is true,
its contents are deleted.
Otherwise, the files in the existing directory remain intact.

Currently all directories in the path leading up to the last one must already exist--this function only creates one subdirectory.

Inlets:

path (Pathname)

delete existing files (Boolean) DEFAULT VALUE: FALSE

Outlets:

status (Error)

status information

Folder: Directory

Get Current Directory

Return the pathname of the "current directory" for the application.

Inlets: none.

Outlets:

dirpath (Pathname)

Folder: Directory

Set Current Directory

Set the "current directory" to the pathname specified.
An error is generated if the directory does not exist.
See also Get Current Directory.

Inlets:

new dir (Pathname) DEFAULT VALUE:

Outlets:

status (Error)

status information

Folder: Directory

Get Path Space Used

Given a path, report the total bytes used for files and directories on that path.
If Subdir is TRUE, subdirectories of the path are searched.

Inlets:

path (Pathname)

subdir (Boolean) DEFAULT VALUE: FALSE

Outlets:

status (Error)

status information

bytes used (Decimal)

Folder: Directory

Get Disk Space

Given a disk drive, report the total bytes used for files and directories on that device.
An example device is: C

Inlets:

disk path (Text)

Outlets:

status (Error)

status information

bytes free (Decimal)

bytes total (Decimal)

Folder: Directory

Delete File

Delete a file given a file name.
This is now obsolete, use Delete Path instead.

Inlets:

filename (Text)

Outlets:

status (Error)

status information

Folder: Directory

Rename File

Rename an existing file.
The file must not be currently open.

Inlets:

oldfilename (Pathname)

newfilename (Pathname)

Outlets:

status (Error)

status information

Folder: Directory

Temp Filename

Generate a unique name for a temporary file, which will normally
be located in the system's temporary directory.
By default the filename will begin with "FVP" and have the
file extension "TMP".
Supplying values for NAME and EXT will override the prefix and
file type, respectively.
These text values must make legitimate file names and types.

Inlets:

name (Text) DEFAULT VALUE: FVP

unique filename's prefix

ext (Text) DEFAULT VALUE: TMP

unique filename's extension

Outlets:

path (Pathname)

Folder: Directory

List All Drives

Get a list of all the disk drives available on this computer.
Each item in the list is of the form "C:\".

Inlets: none.

Outlets:

(List of Pathnames)

Folder: Directory

Disk Type

Returns the kind of disk drive for a path.
The path must be specified in the form "C:\".
(Note that "C" or "C:" will result in a value of 1.)
REMOVABLE: 2,

FIXED: 3,
REMOTE: 4,
CDROM: 5,
RAMDISK: 6,
UNKNOWN: 0,
NO_ROOT_DIR: 1

Inlets:

disk (Pathname)

Outlets:

(Integer)

Folder: Directory

Logical folder

Boolean functions.

Contents of folder:

- And
- Or
- Not

Folder: General

And

Return True if both inputs are True; otherwise return False.

Inlets:

(List of Boolean) VARYING NUMBER OF INLETS DEFAULT VALUE:

Outlets:

a&&b (Boolean)

Folder: Logical

Or

Return True if either inputs are True; otherwise return False.

Inlets:

(List of Boolean) VARYING NUMBER OF INLETS DEFAULT VALUE:

Outlets:

a||b (Boolean)

Folder: Logical

Not

Return the opposite value of the input: True -> False, False->True.

Inlets:

b (Boolean)

Outlets:

!b (Boolean)

Folder: Logical

System folder

Functions for interacting with other processes and applications on the computer.

Contents of folder:

- Launch Application
- Run DOS Command
- Run DOS Commands
- Run DOS Command File
- Open File
- Print File
- Exit
- Find Window
- Send Keys
- GetEnv
- Copy To Clipboard
- Paste From Clipboard
- Get File Object
- Computer Name
- User Name
- Windows Version
- Windows Directories
- Processor Info
- Computer Memory
- List Services

Service Status

Folder: General

Launch Application

Start a DOS or Windows application running in parallel with this application.

Inlets:

app (Text)

Filename of application to lauch, for example: c:\windows\notepad.exe

args (Text) DEFAULT VALUE:

Arguments to the application as if entered on the command line

show (Boolean) DEFAULT VALUE: TRUE

True opens window; False runs application minimized (as an icon).

Outlets:

status (Error)

status information

Folder: System

Run DOS Command

Execute the specified DOS command.

Inlets:

cmd (Text)

a single DOS command line

append (Boolean) DEFAULT VALUE: FALSE

if TRUE and if LOG file supplied, appends to LOG file

log (Pathname) DEFAULT VALUE:

if supplied, writes command output to this file

Outlets:

status (Error)

status information

log (Pathname)

if supplied, writes command output to this file

Folder: System

Run DOS Commands

Execute a list of DOS commands.

Inlets:

cmds (List of Text)

a list of DOS commands to execute, in order

log (<u>Pathname</u>) DEFAULT VALUE:

if supplied, writes command output to this file

append (<u>Boolean</u>) DEFAULT VALUE: FALSE

if TRUE and if LOG file supplied, appends to LOG file

Outlets:

status (<u>Error</u>)

status information

log (<u>Pathname</u>)

Folder: System

Run DOS Command File

Execute the .BAT or .CMD file.

Inlets:

cmd file (<u>Pathname</u>)

a .BAT or .CMD file holding DOS commands

log (<u>Pathname</u>) DEFAULT VALUE:

if supplied, writes command output to this file

append (<u>Boolean</u>) DEFAULT VALUE: FALSE

if TRUE and if LOG file supplied, appends to LOG file

Outlets:

status (<u>Error</u>)

status information

log (<u>Pathname</u>)

Folder: System

Open File

Open a file, using the standard application for that kind of file.

Inlets:

filename (<u>Pathname</u>)

Outlets:

status (<u>Error</u>)

status information

Folder: System

Print File

Print a file, using the standard application for that kind of file.

Inlets:

filename (Pathname)

Outlets:

status (Error)

status information

Folder: System

Exit

Exit the current application, returning an integer status code.

Inlets:

status (Integer) DEFAULT VALUE: 0

Outlets: none.

Folder: System

Find Window

Locate a window given the first part of its name.
Given text, search all active windows and return the window handle of the first window containing the text.
Because it may take some time for applications to bring up certain windows, this function will keep trying to find the window for a certain number of seconds before giving up and returning 0.

Inlets:

title (Text)

Beginning of window title

timeout (Integer) DEFAULT VALUE: 1

how many seconds to keep looking for the window

Outlets:

window (Integer)

Window handle

Folder: System

Send Keys

Send text to a windows application as if it had been typed in.
Each key is represented by one or more characters

To specify most keyboard characters, use the character itself.
If you want to represent more than one character, append each additional character to the one before. For example, to represent the sequence of letters a, b, and c, Text is abc.

The plus (+), caret (^), and percent sign (%) have special meanings. To specify one of these special characters, enter the character inside braces. For example, to specify the plus sign, use {+}.
To send a { character or a } character, use {{} and {}}, respectively.

To specify characters that are not displayed when you press a key (such as Enter or Tab) and other keys that represent actions rather than characters, use the codes shown below:

Key	Code
Backspace	{backspace} or {bs} or {bksp}
Break	{break}
Caps Lock	{capslock}
Clear	{clear}
Del	{delete} or {del}
End	{end}
Enter	{enter} or ~
Esc	{escape} or {esc}
Help	{help}
Home	{home}
Insert	{insert}
Num Lock	{numlock}
Page Down	{pgdn}
Page Up	{pgup}
Print Screen	{prtsc}
Scroll Lock	{scrolllock}
Tab	{tab}
Arrows...	{left}, {right}, {up}, {down}
F1-F16	{f1}-{f16}

To specify keys combined with any combination of Shift, Ctrl, and Alt, precede the regular key code with one or more of these codes:

Key	Code
Shift	+
Ctrl	^
Alt	%

To specify that Shift, Ctrl, and/or Alt should be held down while several keys are pressed, enclose the keys in parentheses. For example, to hold down the Shift key while pressing E then C, use +(EC). To hold down Shift while pressing E, followed by C without the Shift key, use +EC.   To specify repeating keys, use the form {key number} where there is always a space between key and number. For example, {left 42} means press the left arrow key 42 times; {x 10} means press the character x 10 times.

Inlets:

window (Integer) DEFAULT VALUE: 0

Window handle of window to send keys to

keys (Text)

Keys to send

Outlets:

status (Error)

Status of operation

Folder: System

GetEnv

Get the value of a system environment variable.

Inlets:

name (Text)

Outlets:

transl (Text)

Folder: System

Copy To Clipboard

Put a value into the Windows clipboard.

Inlets: none.

Outlets: none.

Folder: System

Paste From Clipboard

Get a value from the Windows clipboard.

Inlets: none.

Outlets: none.

Folder: System

Get File Object

Get an OLE object associated with a file.

Inlets:

filename (Text)

OLE ID (Text) DEFAULT VALUE:

Outlets:

Object (***any***)

Folder: System

Computer Name

Get the name of this computer.

Inlets: none.

Outlets:

(Text)

Folder: System

User Name

Get the name of the user currently logged in.

Inlets: none.

Outlets:

(Text)

Folder: System

Windows Version

Return whether running Windows NT.
Get the major and minor release numbers and the build number for the operating system.

Inlets: none.

Outlets:

NT (Boolean)

major (Integer)

minor (Integer)

build (Integer)

patch (Text)

Folder: System

Windows Directories

Get the directories where the Windows operating system resides, typically C:\WINDOWS and C:\WINDOWS\SYSTEM.

Inlets: none.

Outlets:

windows (Pathname)

system (Pathname)

Folder: System

Processor Info

Get the kind of processor used in this computer,
and how many there are.

Inlets: none.

Outlets:

kind (Text)

num (Integer)

Folder: System

Computer Memory

Return how much physical memory there is on this computer,
and how much is unused..

Also return how much of the page file could be used,
and how much is currently available.

Inlets: none.

Outlets:

mem free (Integer)

(bytes)

mem total (Integer)

(bytes)

page free (Integer)

(pages)

page total (Integer)

(pages)

Folder: System

List Services

Return a list of the Service names on this computer.

Inlets: none.

Outlets:

err (Error)

names (List of Text)

Folder: System

Service Status

Get information about the current state of a Service on this computer.
State values: 1: Stopped, 2: Start pending, 3: Stop pending, 4: Running,
5: Continue pending, 6: Pause pending, 7: Paused

Inlets:

name (Text)

Outlets:

err (Error)

state (Integer)

sys code (Integer)

serv code (Integer)

Folder: System

Time folder

Functions for manipulating date/time values.

Contents of folder:

- Wait
- Time Now
- Make Time
- Time Parts
- Format Time
- Time to Text Alias
- Text to Time Alias
- Add Time
- Subtract Time
- Time Difference
- Earlier Than
- Later Than
- Earlier or Same Time
- Later or Same Time
- Same Time
- Different Time

Folder: General

Wait

Wait for the specified time, doing nothing. The wait, combining hours, minutes, and seconds, may be at most 1 million seconds.

Inlets:

hours (Integer) DEFAULT VALUE: 0

minutes (Integer) DEFAULT VALUE: 0

seconds (Integer) DEFAULT VALUE: 0

Outlets:

status (Boolean)

Folder: Time

Time Now

Return the current date/time value.

Inlets: none.

Outlets:

time (Time)

Folder: Time

Make Time

Create a date/time value from the integer values for each of its parts.

Inlets:

sec (Integer)

0-59 seconds after the minute

min (Integer)

0-59 minutes after the hour

hour (Integer)

0-23 hours after midnight

date (Integer)

1-31 date of the month

month (Integer)

1-12 month of the year

year (Integer)

1970-2036

Outlets:

time (Time)

Folder: Time

Time Parts

Given a date/time value, return the values for each of its parts.

Inlets:

time (Time)

Outlets:

time (Time)

sec (Integer)

0-59 seconds after the minute

min (Integer)

0-59 minutes after the hour

hour (Integer)

0-23 hours after midnight

date (Integer)

1-31 date of the month

month (Integer)

1-12 month of the year (January is 1)

year (Integer)

1970-2036

weekday (Integer)

1-7 day of the week (Sunday is 1)

yearday (Integer)

1-366 day of the year (January 1st is 1)

DST (Boolean)

TRUE if Daylight Saving Time is in effect

Folder: Time

Format Time

Convert a date/time value to text.
FORMAT determines how the value will be expressed as text.
Each occurrence of %x in FORMAT is replaced by some time value, according to the following rules:

Any %A is replaced by the name of the weekday
Any %B is replaced by the name of the month
Any %m is replaced by the month as an integer, 1-12
Any %d is replaced by the date of the month, 1-31
Any %y is replaced by the year, 0-99
Any %Y is replaced by the year, including century
Any %H is replaced by the hour, 0-23
Any %I is replaced by the hour, 1-12

Any %M is replaced by the minute
Any %S is replaced by the second
Any %p is replaced by AM or PM as appropriate

Inlets:

format (Text) DEFAULT VALUE: %X %x

time (Time)

Outlets:

string (Text)

Folder: Time

Add Time

Calculate a new date/time value which is some number of seconds, minutes, hours, or days later than a given time.

Inlets:

time (Time)

secs (Integer) DEFAULT VALUE: 0

mins (Integer) DEFAULT VALUE: 0

hours (Integer) DEFAULT VALUE: 0

days (Integer) DEFAULT VALUE: 0

Outlets:

time (Time)

Folder: Time

Subtract Time

Calculate a new date/time value which is some number of seconds, minutes, hours, or days earlier than a given time.

Inlets:

time (Time)

secs (Integer) DEFAULT VALUE: 0

mins (Integer) DEFAULT VALUE: 0

hours (Integer) DEFAULT VALUE: 0

days (Integer) DEFAULT VALUE: 0

Outlets:

time (Time)

Folder: Time

Time Difference

Return the difference between two date/time values in seconds, minutes, hours, and days.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

secs (Integer)

0-59

mins (Integer)

0-59

hours (Integer)

0-23

days (Integer)

>= 0

Folder: Time

Earlier Than

Return TRUE if the T1 date/time comes before T2.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1<t2 (Boolean)

Folder: Time

Later Than

Return TRUE if the T1 date/time comes after T2.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1>t2 (Boolean)

Folder: Time

Earlier or Same Time

Return TRUE if the T1 date/time comes before T2, or if they are the same date/time.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1<=t2 (Boolean)

Folder: Time

Later or Same Time

Return TRUE if the T1 date/time comes after T2, or if they are the same date/time.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1>=t2 (Boolean)

Folder: Time

Same Time

Return TRUE if T1 is the same date/time value as T2.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1==t2 (Boolean)

Folder: Time

Different Time

Return TRUE if T1 is not the same date/time as T2.

Inlets:

t1 (Time)

t2 (Time)

Outlets:

t1!=t2 (Boolean)

Folder: Time

Registry folder

Functions for manipulating the Windows, or Windows NT Registry

Contents of folder:

- Registry Get Text
- Registry Get Integer
- Registry Set Text
- Registry Set Integer
- Registry SubKey Names
- Registry Value Names
- Registry Delete Key
- Registry Delete Value
- Registry Key Exists
- Read Profile
- Write Profile

Folder: General

Registry Get Text

Get the text value of a key in the registry. A registry key
is a text value that looks like a file directory path.
The key should always begin with one of the following:

HKEY_CLASSES_ROOT\	or	HKCR\
HKEY_CURRENT_USER\	or	HKCU\
HKEY_LOCAL_MACHINE\	or	HKLM\
HKEY_USERS\	or	HKU\

Each registry key may have several named values; each always has
one named value whose name is a zero-length text string.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

Outlets:

err (Error)

val (Text)

Folder: Registry

Registry Get Integer

Get the integer value of a key in the registry. A registry key
is a text value that looks like a file directory path.
The key should always begin with one of the following:

HKEY_CLASSES_ROOT\	or	HKCR\
HKEY_CURRENT_USER\	or	HKCU\
HKEY_LOCAL_MACHINE\	or	HKLM\
HKEY_USERS\	or	HKU\

Each registry key may have several named values; each always has
one named value whose name is a zero-length text string.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

Outlets:

err (Error)

val (Integer)

Folder: Registry

Registry Set Text

Change the text value of a key in the registry, perhaps creating the key
if necessary. A registry key
is a text value that looks like a file directory path.
The key should always begin with one of the following:

HKEY_CLASSES_ROOT\	or	HKCR\
HKEY_CURRENT_USER\	or	HKCU\
HKEY_LOCAL_MACHINE\	or	HKLM\
HKEY_USERS\	or	HKU\

Each registry key may have several named values; each always has
one named value whose name is a zero-length text string.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

val (Text)

Outlets:

err (Error)

Folder: Registry

Registry Set Integer

Change the integer value of a key in the registry, perhaps creating the key if necessary. A registry key
is a text value that looks like a file directory path.
The key should always begin with one of the following:

HKEY_CLASSES_ROOT\	or	HKCR\
HKEY_CURRENT_USER\	or	HKCU\
HKEY_LOCAL_MACHINE\	or	HKLM\
HKEY_USERS\	or	HKU\

Each registry key may have several named values; each always has one named value whose name is a zero-length text string.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

val (Integer)

Outlets:

err (Error)

Folder: Registry

Registry SubKey Names

Return a list of key names which are a part of the given key.

Inlets:

key (Text)

Outlets:

err (Error)

subkeys (List of Text)

Folder: Registry

Registry Value Names

Return a list of names for values for the given key.
The default value name is included in this list as the empty string (a name with no characters in it).

Inlets:

key (Text)

Outlets:

err (Error)

names (List of Text)

Folder: Registry

Registry Delete Key

Delete the given key and any values and subkeys that are a part of it.

Inlets:

path (Text)

Outlets:

err (Error)

Folder: Registry

Registry Delete Value

Remove a particular named value from a particular key in the registry.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

Outlets:

err (Error)

Folder: Registry

Registry Key Exists

Returns TRUE if the given key and value name exists in the registry.
If it exists and has a value, it also returns the type of value:
0: no value
1: text string
2: text string
4: integer
Other value types are not supported at this time.

Inlets:

key (Text)

name (Text) DEFAULT VALUE:

Outlets:

? (Boolean)

kind (Integer)

Folder: Registry

Read Profile

Read a value from an .INI file.

Inlets:

Section (Text)

Entry (Text)

Default (Text) DEFAULT VALUE:

BufferSize (Integer) DEFAULT VALUE: 255

INI File (Pathname)

Outlets:

StringLength (Integer)

String (Text)

Folder: Registry

Write Profile

Write a value to a .INI file.

Inlets:

Section (Text)

Entry (Text)

String (Text)

INI File (Pathname)

Outlets:

Status (Boolean)

Folder: Registry

Error folder

Functions for creating and examining error values.

Contents of folder:

- Error!
- Error Parts
- Ignore Error
- Error Match?
- Success!
- Display Error

Folder: General

Error!

Create an error object--if the value is unused it also signals the error so that error handlers in calling functions can handle it.

Inlets:

facility (Text) DEFAULT VALUE:

error facility name

name (Text) DEFAULT VALUE:

name which identifies the error

code (Integer) DEFAULT VALUE: 0

external error specific code

msg (Text) DEFAULT VALUE:

a message describing the error

help file (Pathname) DEFAULT VALUE:

help file which contains help on this error

help ctxt (Text) DEFAULT VALUE:

key into help file

Outlets:

error (Error)

Folder: Error

Error Parts

Get the text message and code and other information about an error object.

Inlets:

error (Error)

Outlets:

error? (Boolean)

True if this contains an error status

facility (Text)

error facility name

name (Text)

name which identifies the error

code (Integer)

external error specific code

msg (Text)

a message describing the error

help file (Pathname)

help file which contains help on this error

help ctxt (Text)

key into help file

Folder: Error

Ignore Error

Avoid a signal from an unused error value.

Inlets:

error (Error)

Outlets: none.

Folder: Error

Error Match?

Determine if an error value's facility name and error name match
some given names.

Inlets:

error (Error)

error value

facility name (Text) DEFAULT VALUE:

error facility name

error name (Text) DEFAULT VALUE:

name which identifies the error

Outlets:

match? (Boolean)

true if the error matches

Folder: Error

Success!

Produce an error value that does not cause a signal because it indicates success rather than failure.

Inlets: none.

Outlets:

status (Error)

Folder: Error

Display Error

Bring up a dialog box showing the text of the error.

Inlets:

error (Error)

Outlets: none.

Folder: Error

DDE folder

Functions for conducting DDE conversations on Windows.

Contents of folder:

- DDE Connect
- DDE Disconnect
- DDE Execute
- Poke DDE Data
- Request DDE Data
- Clipboard Format
- Request DDE Text
- Poke DDE Text

Folder: General

DDE Connect

Attempts to initiate a DDE conversation with the specified service and topic. Returns a conversation id (conv_id) which is nonzero if a conversation has been established successfully.
See your application's documentation for valid DDE services and topics.

Inlets:

service (Text)

topic (Text)

Outlets:

conv_id (Integer)

DDE conversation ID, to be passed to other DDE functions

status (Error)

status information

Folder: DDE

DDE Disconnect

Ends the DDE conversation identified by the conv_id. See DDE Connect for information on starting a conversation and obtaining a conv_id.

Inlets:

conv_id (Integer)

DDE conversation ID

Outlets:

status (Error)

status information

Folder: DDE

DDE Execute

Sends a command to the service identified by the conv_id.
See the application's documentation for acceptable commands.
An optional timeout value (in milliseconds) can be specified,
to control how long to keep trying to execute the command.

Inlets:

conv_id (Integer)

DDE conversation ID

command (Text)

timeout (Integer) DEFAULT VALUE: 30000

how long to keep trying, in milliseconds

Outlets:

conv_id (Integer)

status (Error)

status information

Folder: DDE

Poke DDE Data

Sends a value to the specified item.
See your application documentation for valid items.

Inlets:

conv_id (Integer)

DDE conversation ID

item (Text)

data (DataObject)

data_length (Integer) DEFAULT VALUE: 0

data_format (Integer) DEFAULT VALUE: 1

timeout (Integer) DEFAULT VALUE: 30000

how long to keep trying, in milliseconds

Outlets:

conv_id (Integer)

status (Error)

status information

Folder: DDE

Request DDE Data

Request arbitrary data in a DDE conversation.
Retrieves the value of the specified item.
See yor application documentation for valid items.

Inlets:

conv_id (Integer)

DDE conversation ID

item (Text)

data_format (Integer) DEFAULT VALUE: 1

timeout (Integer) DEFAULT VALUE: 30000

how long to keep trying, in milliseconds

Outlets:

conv_id (Integer)

data (DataObject)

data_length (Integer)

status (Error)

status information

Folder: DDE

Clipboard Format

Determine if a particular data format is currently available on the Windows clipboard.

Inlets:

Format Name (Text)

Outlets:

Format ID (Integer)

Folder: DDE

Request DDE Text

Request an item of text in a DDE conversation.

Inlets:

conv_id (Integer)

DDE conversation ID

item (Text)

timeout (Integer) DEFAULT VALUE: 30000

how long to keep trying, in milliseconds

Outlets:

conv_id (Integer)

text (Text)

status (Error)

status information

Folder: DDE

Poke DDE Text

Send an item of text in a DDE conversation.

Inlets:

conv_id (Integer)

DDE conversation ID

item (Text)

text (Text)

timeout (Integer) DEFAULT VALUE: 30000

how long to keep trying, in milliseconds

Outlets:

conv_id (Integer)

DDE conversation ID

status (Error)

status information

Folder: DDE

Exp & Trig folder

Exponential and Trigonometric functions.

Contents of folder:

- Cosine
- Sine
- Tangent
- ArcCosine
- ArcSine
- ArcTangent
- Hyperbolic Cosine
- Hyperbolic Sine
- Hyperbolic Tangent
- Exponentiate
- Logarithm

Folder: General

Cosine

Given an angle in radians, compute the Cosine.

Inlets:

f (Decimal)

Outlets:

cos(f) (Decimal)

Folder: Exp & Trig

Sine

Given an angle in radians, compute the Sine.

Inlets:

f (Decimal)

Outlets:

sin(f) (Decimal)

Folder: Exp & Trig

Tangent

Given an angle in radians, compute the Tangent.

Inlets:

f (Decimal)

Outlets:

tan(f) (Decimal)

Folder: Exp & Trig

ArcCosine

Given an angle in radians, compute the ArcCosine.

Inlets:

f (Decimal)

Outlets:

acos(f) (Decimal)

Folder: Exp & Trig

ArcSine

Given an angle in radians, compute the ArcSine.

Inlets:

f (Decimal)

Outlets:

asin(f) (Decimal)

Folder: Exp & Trig

ArcTangent

Given an angle in radians, compute the ArcTangent.

Inlets:

f (Decimal)

Outlets:

atan(f) (Decimal)

Folder: Exp & Trig

Hyperbolic Cosine

Given an angle in radians, compute the Hyperbolic Cosine.

Inlets:

f (Decimal)

Outlets:

cosh(f) (Decimal)

Folder: Exp & Trig

Hyperbolic Sine

Given an angle in radians, compute the Hyperbolic Sine.

Inlets:

f (Decimal)

Outlets:

sinh(f) (Decimal)

Folder: Exp & Trig

Hyperbolic Tangent

Given an angle in radians, compute the Hyperbolic Tangent.

Inlets:

f (Decimal)

Outlets:

tanh(f) (Decimal)

Folder: Exp & Trig

Exponentiate

Return e (the mathematical constant) raised to the f power.

Inlets:

f (Decimal)

Outlets:

e^f (Decimal)

Folder: Exp & Trig

Logarithm

Compute the logarithm of a decimal number f.

Inlets:

f (Decimal)

Outlets:

ln(f) (Decimal)

Folder: Exp & Trig

Conversions folder

Data type conversions

Contents of folder:

- Integer To Decimal
- Decimal To Integer
- Round Decimal To Integer
- Text To Integer
- Text To Decimal
- Integer To Text
- Decimal To Text
- Boolean To Text
- Text To Boolean
- Write List As Text Alias
- Read List Flexibly Alias
- Text to Path Alias
- Boolean To Integer
- Integer To Boolean
- Time to Text
- Text to Time
- Is Text an Integer
- Is Text a Decimal

Folder: General

Integer To Decimal

Convert integer to decimal.

Inlets:

i (Integer)

Outlets:

f (Decimal)

Folder: Conversions

This Function acts as an automatic conversion from Integer to Decimal

---

Decimal To Integer

Convert Decimal to Integer.

Inlets:

f (Decimal)

Outlets:

i (Integer)

Folder: Conversions

Round Decimal To Integer

Convert decimal to integer; round up fraction.

Inlets:

f (Decimal)

Outlets:

i (Integer)

Folder: Conversions

This Function acts as an automatic conversion from Decimal to Integer

---

Text To Integer

Convert Text to Integer.

Inlets:

s (Text)

Outlets:

i (Integer)

Folder: Conversions

This Function acts as an automatic conversion from Text to Integer

---

Text To Decimal

Convert Text To Decimal

Inlets:

s (Text)

Outlets:

f (Decimal)

Folder: Conversions

This Function acts as an automatic conversion from Text to Decimal

---

Integer To Text

Convert Integer To Text.

Inlets:

i (Integer)

Outlets:

s (Text)

Folder: **Conversions**

This Function acts as an automatic conversion from Integer to Text

---

Decimal To Text

Convert Decimal To Text.

Inlets:

f (Decimal)

Outlets:

s (Text)

Folder: Conversions

This Function acts as an automatic conversion from Decimal to Text

---

Boolean To Text

Convert Boolean To Text.

Inlets:

b (Boolean)

Outlets:

s (Text)

Folder: Conversions

This Function acts as an automatic conversion from Boolean to Text

---

Text To Boolean

Convert Text To Boolean.

Inlets:

s (Text)

Outlets:

b (Boolean)

Folder: Conversions

This Function acts as an automatic conversion from Text to Boolean

---

Boolean To Integer

Convert Boolean To Integer (0 if FALSE, 1 if TRUE)

Inlets:

b (Boolean)

Outlets:

i (Integer)

Folder: Conversions

Integer To Boolean

Convert Integer To Boolean (FALSE if 0, TRUE otherwise)

Inlets:

i (Integer)

Outlets:

b (Boolean)

Folder: Conversions

Time to Text

Convert a time value to text using a standard format.

Inlets:

time (Time)

Outlets:

string (Text)

Folder: Conversions

This Function acts as an automatic conversion from Time to Text

Text to Time

Convert a text string into a time value using a standard format.

Inlets:

text (Text)

Outlets:

time (Time)

Folder: Conversions

This Function acts as an automatic conversion from Text to Time

Is Text an Integer

TRUE if the given text string represents an integer.
Initial whitespace and characters after the integer are ignored.

Inlets:

text (Text)

Outlets:

i? (Boolean)

Folder: Conversions

Is Text a Decimal

TRUE if the given text string represents a decimal number.
Initial whitespace and characters after the number are ignored.

Inlets:

text (Text)

Outlets:

f? (Boolean)

Folder: Conversions

Language folder

Functions for expressing concepts in the visual language. One can also find these and others on the Insert menu, or on the pop-up menu when pressing the right button of the mouse on the background of a flowgram's canvas.

Contents of folder:

- New Comment
- New Constant
- New Form
- New Package
- New Pick One
- New Repeat
- New Stop Repeat
- New Error Handler
- New Get OLE Property
- New Set OLE Property

Folder: General

New Comment

Insert a new comment into a flowgram.

Inlets: none.

Outlets: none.

Folder: Language

New Constant

Insert a new constant into a flowgram. The constant will not have an initial value or even an initial data type.

Inlets: none.

Outlets: none.

Folder: Language

New Form

Insert a new modal dialog form into a flowgram.

One can add controls to a form, such as text fields, list boxes, and buttons.

Inlets: none.

Outlets: none.

Folder: Language

New Package

Insert a new package into a flowgram, to help simplify diagrams by hiding details.

Inlets: none.

Outlets: none.

Folder: Language

New Pick One

Insert a new Pick One into a flowgram.
Depending on the input value, it will perform exactly one of the flowgrams contained inside this Pick One. The input value must be an integer, a decimal, a text value, or a boolean (TRUE/FALSE).

Inlets: none.

Outlets: none.

Folder: Language

New Repeat

Insert a new Repeat into a flowgram.
This will repeatedly execute the body of the Repeat, according to the input value. A positive integer will cause the body to be performed that number of times; a list will cause the body to be performed once for each item of the list. One can terminate the iteration prematurely by calling a Stop in Repeat with a value of TRUE.

Inlets: none.

Outlets: none.

Folder: Language

New Stop Repeat

Insert a Stop into the body of a Repeat.
If any Stop inside a Repeat gets passed a TRUE value,

the Repeat loop will stop.

Inlets: none.

Outlets: none.

Folder: Language

New Error Handler

Insert an error handler into a flowgram.
There can be at most one error handler in any flowgram.
The body of the error handler is executed whenever an error occurs in the flowgram or in any other function called by that flowgram. The error handler's values take the place of the flowgram's output connector values.

Inlets: none.

Outlets: none.

Folder: Language

New Get OLE Property

Insert a function into a flowgram that gets the value of a property of an OLE object. The label names the property.

Inlets: none.

Outlets: none.

Folder: Language

New Set OLE Property

Insert a function into a flowgram that sets the value of a property of an OLE object. The label names the property.

Inlets: none.

Outlets: none.

Folder: Language

Convert Interface

Convert a COM Interface object to a different one (using QueryInterface). This function may be required in order to apply certain methods to a COM object. Refer to the documentation for the type library that defines the COM object.

Inlets:

object (COM object)

Outlets:

object (QueryInterface *object)*

status (Error)

Folder: Language

Select Value

Output one inlet value or another, based upon the value of a third inlet.

Inlets:

? (Boolean)

This inlet determines whether X or Y is passed through

X (***any***)

The value passed through if ? is TRUE. The same Type as *Y*

Y (***any***)

The value passed through if ? is FALSE. The same Type as *X*

Outlets:

(***any***)

The value of either *X* or *Y*, depending on the value of ?. The same Type as *X.*

Folder: Language