

os function group	DSEX extended os function group	Functionality
chdir(d)	chdir(d)	Change the working directory to d (the directory path).
chmod(f,p)	chmod(f,p)	Changes permissions for file f using UNIX permission bits (supplied as a string, e.g., "777").
clock()	clock()	Returns the time used by the CPU (in seconds).
	currentdir()	Returns the path to the current directory.
cwd()	cwd()	Returns the path to the current directory.
date([f,t])	date([f,t])	Returns the current date and time as a string with format f; default is current date/time with format DD/MM/CCYY HH:MI:SS. Argument t is a value returned by the os.time() function.
difftime(t1,t2)	difftime(t1,t2)	Returns the difference between two values returned by the os.time() function.
	dir(p)	An iterator function for the contents in directory p. Returns a string indexed table that includes indices type, name, & size for each file or subdirectory. for entry in os.dir(p) do ... end
	dirent(f)	Returns a string indexed table that includes indices type and size for the file f.
	environ	Returns a string indexed table where the indices are the names of environment variables and the values are the corresponding environment variable setting.
execute(c)	execute(c)	Executes the command c. Waits until the command completes and returns the exit code from the command.
exit()	exit()	Terminates the program.
getenv(e)	getenv(e)	Returns the value of the environment variable e.
is_dir(p)	is_dir(p)	Returns true if p is a directory.
is_file(f)	is_file(f)	Returns true if f is a directory.
mkdir(p)	mkdir(p)	Creates the directory p.
readdir(p)	readdir(p)	Returns a numerically indexed table where each entry is the name of a file or subdirectory in directory p.
remove(f)	remove(f)	Deletes file f. On error, returns nil and error description.
rename(of,nf)	rename(of,nf)	Renames file of to nf. On error, returns nil and error description.
rmdir(p)	rmdir(p)	Removes directory p. On error, returns nil and error description.
	setenv(e,v)	Sets the value of environment variable e to v. If v is nil, deletes environment variable; if variable doesn't exist, returns nil and error description.
setlocale(s[,c])	setlocale(s[,c])	Sets the local set by string s for category c. Categories: "all", "collate", "ctype", "monetary", "numeric", "time"; default "all".
	sleep(d[,u])	Suspends program execution for duration d in units u. Duration may be decimal; default unit is seconds.
	spawn(p[,a])	Creates a child process for program p. If specified, a is one of the following: <ul style="list-style-type: none"> • A numerically indexed table of command line arguments • A string indexed table of environment variables • stdin, stdout, stderr file handles Returns a user object (proc) that implements the following function exitcode = proc:wait() that waits for the child process to complete before completing; exitcode is the value returned by the child process.
time([t])	time([t])	Returns a system dependent number representing a datetime described by the string indexed table t. Table t must have the index entries year, month, day and may have entries for hour, min, sec, isdst(true=daylight savings time).
tmpname()	tmpname()	Returns a random file name (does not create the file).

Note that when using the DSEX function setenv, the operating system's environment variables are not modified. To extract the value of variables set by this function you must use the os.spawn function as illustrated in the following statement, where ... is the name of the environment variable.

```
proc=os.spawn("datascript", {"-e", "print(os.getenv('...'))"}); proc:wait()
```

DSEX extended io function group	Functionality
f.lock(mode,offset,length)	Lock or unlock a file or a portion of a file associated with the file handle f. Mode: "r", read; "w", write; "u", unlock. Offset and length are optional positions in the file.
f.unlock(offset,length)	Unlocks a file or portion of a file associated with the file handle f. Offset and length are optional positions in the file.
io.lock(f,mode,offset,length)	Lock or unlock the file f or a portion of the file f. Mode: "r", read; "w", write; "u", unlock. Offset and length are optional positions in the file.
io.unlock(f,offset,length)	Unlocks a file or portion of the file f. Offset and length are optional positions in the file.