Binary Variables

/bread

Added in 1.9.0

/bread [-ta] <filename> <S> <N> <&binvar>

Reads <N> bytes starting at byte position <S> in the file and stores the result in &binvar.

Switches

-t - reads data until the next $crlf or $feof.
-a - Disables UTF-8 encoding of characters in the range 0-255, as long as the line contains no characters > 255.

Parameters

<filename> - The binary file to read.
<S> - The byte position in the file to start reading from.
<N> - Number of bytes to read.
<&binvar> - The &binvar to read the data into.

Example

```
;noop $example(FileA, FileB)
alias example {
   ;Read the whole file into a binary variable
   bread $qt($1) 0 $file($1).size &tempFile

   ;Write the bytes form the binary variable to a file
   bwrite $qt($2) 0 -1 &tempFile
}
```

/bwrite

Added in 1.9.0

/bwrite [-tac] <filename> <S> [N] <text|%var|&binvar>

Writes [N] bytes from <text|%var|&binvar> to the file starting at byte position <S> or 0, any existing information at this position is overwritten.

Switches

-t - Treat <text|%var|&binvar> as plain text.
-a - Disables UTF-8 encoding of characters in the range 0-255, as long as the text contains no characters > 255.
-c - Chops the file at the end of the copied bytes.

Parameters

<filename> - File to modify.
<S> - Byte position in the file to start writing to. (zero based)
[N] - Byte count from <text|%var|&binvar> to write.
<text|%var|&binvar> - Text/%variable/&binvar to write to file.

Example

```
alias example {
  ;Write some text to a file at beginning of the file
  /bwrite file.txt 0 hello there!
```

```
  ;Read the binary data into binary variable &tempfile
  /bread $qt(file.txt) 0 $file(file.txt).size &tempfile

  ;Print the binary variable data as text, outputs 'Hello there!'
  echo -ag $bvar(&tempfile, 1-).text

  ;Replace "there!" with "world!"
  /bwrite file.txt 6 world!

  ;Read the binary data into binary variable &tempfile
  /bread $qt(file.txt) 0 $file(file.txt).size &tempfile

  ;Print the binary variable data as text, outputs 'Hello world!'
  echo -ag $bvar(&tempfile, 1-).text
}
```

/bset

Added in 1.9.0

/bset [-tac] <&binvar> <N> <asciivalue> [asciivalue ... asciivalue]

Sets the <N>th byte in binary variable &binvar to the specified ascii value.

If N = -1, the data is added to the end of the variable.

If the &binvar docent exists it is created.
If <N> is larger than the size of &binvar it will be zero filled up to <N> bytes.
If you specify multiple ASCII values, they are copied to successive positions after byte position N.

Switches

-t - Treat values a plain text.
-a - Disables UTF-8 encoding of characters in the range 0-255, as long as the text contains no characters > 255.
-c - Chops the &binvar at the end of the copied bytes.

Parameters

<&binvar> - The &binvar to modify.
<N> - The byte to modify. (one based)
<asciivalue> - The ASCII value to insert.

Example

```
alias example {
  ; Create a binary variable set it to "This is fun!"
  bset -t &Example 1 This is fun!

  ; Print out the content of the variable
  echo -a $bvar(&Example, 1-).text
}
```

/bunset

Added in 1.9.0

/bunset <&binvar> [&binvar ... &binvar]

Unsets the specified list of &binvars.

Parameters

<&binvar> - The &binvar to unset.

[&binvar ... &binvar] - Additional &binvars to unset.

/bcopy

Added in 1.9.0

/bcopy [-zc] <&binvar> <N> <&binvar> <S> <M>

Copies <M> bytes from position <S> in the second &binvar to the first &binvar at position <N>.
This can also be used to copy overlapping parts of a &binvar to itself.

Switches

-z - The bytes in the second &binvar that is copied are zero-filled after the copy.
-c - The first &binvar is chopped to <N> + <M>.

Parameters

<&binvar> - Target &binvar to copy to.
<N> - Target position in the first &binvar to copy to. (If N = -1, bytes are appended to the destination &binvar)
<&binvar> - Source &binvar to copy from.
<S> - Source position to copy from.
<M> - Number of bytes to copy.

Example

```
alias /example {
  ;Create a binary variable and assign it some text
  bset -t &example 1 This is a test!

  ;Copy from 'example' from the 11th byte 10 bytes onward
  ;Zero-fill the part that was copied
  bcopy -z &example2 1 &example 11 10

  ;Print out &example's content (up to the first null)
  echo -a $bvar(&example, 1-).text

  ;Print out &example2's content
  echo -a $bvar(&example2, 1-).text
}
```

/breplace

Added in 1.9.0

/breplace <&binvar> <oldvalue> <newvalue> [oldvalue newvalue...]

Replaces all matching ASCII values in &binvar with new values.

Multiple values can be replaced by adding more old/new parameters.

Parameters

<&binvar>- The &binvar to modify.
<oldvalue> - ASCII value to replace.
<newvalue> - ASCII value to insert.

Example

```
alias example {
  ;Create a binary variable set it to "Hello World"
  bset -t &Example 1 Hello World

  ;Replace e (ASCII value 101) with 3 (ASCII value 51)
```

```
  breplace &Example 101 51

  ;Echo our new string
  echo -a $bvar(&Example,1,$bvar(&Example,0)).text
}
```

/btrunc

Added in 1.9.0

/btrunc <filename> <bytes>

Truncates/extends a file to the specified length.

Parameters

<filename> - The file to truncate.
<bytes> - Number of bytes to truncate/extend to.

Example

```
alias example {
  ;Create variable %temp and add some data.
  /var %temp = Hello! World

  ;Write to variable %temp's content.
  /bwrite Example 0 $len(%temp) %temp

  ;Truncate the file down to 6 bytes.
  /btrunc Example 6

  ;Read the file into a variable.
  /bread Example 0 $file(Example).size &Example

  ;Print out the variable's content.
  /echo -a $bvar(&Example,1,$bvar(&Example,0)).text

  ;Delete the file.
  /remove Example
}
```

$bvar

Added in 1.9.0

$bvar(&binvar,N,M)

Returns M ASCII values from a &binvar starting from the Nth byte.

N-N2 can be used to get a range of ASCII values.
N- can be used to get all ASCII values from position N.

Parameters

&binvar - The &binvar to use.
N - Position to start retrieving bytes.
M - Numbers of bytes to get.

Properties

.text - Returns plain text instead of ASCII values.
.word - Outputs decimal value of a 2-byte word, seeing bytes in little-endian byte order (low value first, (unsigned 16 bit)
.nword - Outputs decimal value of a 2-byte word, seeing bytes in big-endian byte order (high value first, unsigned 16 bit)
.sword - Outputs decimal value of a 2-byte word, seeing bytes in little-endian byte order (low value first, signed 16 bit, **AdiIRC only**)

.nsword - Outputs decimal value of a 2-byte word, seeing bytes in big-endian byte order (high value first, signed 16 bit, **AdiIRC only**)

.long - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in little-endian byte order (low value first, unsigned 32 bit)
.nlong - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in big-endian byte order (high value first. unsigned 32 bit)
.slong - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in little-endian byte order (low value first, signed 32 bit, **AdiIRC only**)
.nslong - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in big-endian byte order (high value first. signed 32 bit **AdiIRC only**)

.uint64 - Outputs decimal value of a 8-byte dword (unsigned long), seeing bytes in little-endian byte order (low value first, unsigned 64 bit **AdiIRC only**)
.nuint64 - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in big-endian byte order (high value first. unsigned 64 bit **AdiIRC only**)
.sint64 - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in little-endian byte order (low value first, signed 64 bit, **AdiIRC only**)
.nsint64 - Outputs decimal value of a 4-byte dword (unsigned long), seeing bytes in big-endian byte order (high value first. signed 64 bit **AdiIRC only**)

Example

```
; Returns the length of the binary variable.
//echo -ag $bvar(&binvar,0)

; Returns ASCII value at position N.
//echo -ag $bvar(&binvar,1)

; Returns ASCII values from 5 to 8.
//echo -ag $bvar(&binvar,5,3)

; Returns ASCII values from 5 to end.
//echo -ag $bvar(&binvar,5-)

; Returns plain text from 5 to 8 up to the first zero character.
//echo -ag $bvar(&binvar,5,3).text

; Returns &binvar if the binvar exists.
//echo -ag $bvar(&binvar)
```

$bfind

Added in 1.9.0

$bfind(&binvar, N, M, [name])

Searches a &binvar for a matching value.

The search is case-insensitive.

Parameters

&binvar - The &binvar to search.
N - The position of the &binvar to start the search.
M - Numeric ASCII character(s) or text to search.
[name] - Name to store in $regml for the result(s) when using the .regex property. (optional)

Properties

.text - Force a text search if the search text (M) is a number.
.textcs - Search case-sensitive.
.regex - Performs a regular expression search for the pattern **M**.

Example

```
; Finds "test" starting from position 1.
//echo -a $bfind(&test, 1, test)

; Finds character 32 (a space) from position 5.
//echo -a $bfind(&test, 5, 32)

; Finds WAV from position 1.
//echo -a $bfind(&test, 1, 87 65 86)
```