BBBS v3.14.95 Zebra 5, Script Language

Kim Heino, Tapani T. Salmi, Kai Lunde, and Stig Johansen

COLLABORATORS

	TITLE : BBBS v3.14.95 Zebra 5, Script Language	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Kim Heino, Tapani T. Salmi, Kai Lunde, and Stig Johansen	January 6, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BBBS v3.14.95 Zebra 5, Script Language

1.1 Table of Contents

1.2 What is BZ?

BBBS can run BZ49 machine language code. You can compile BZ language source code files to BZ assembly language or to BZ49 machine language code with BZC program.

Files named *.bz are BZ source codes, *.bzh are header files and *.bzb are compiled binary files. *.bzs are assembly language source code files.

You can use any normal text editor to produce BZ file, as long as its output is normal ASCII text. Like in all major languages, case is important. Whitespace (such as the space, the tab the Carriage Return or the Line Feed character) is not important, but using them makes everything much cleaner to you, the programmer. The compiler does not care where you place items, so that you can arrange the program as you see fit.

1.3 Language Syntax

BZ language is based on functions. Executing the program starts from the main function and returning from it stops the program.

```
Syntax  function foo2($buz) \{
          command1();
          ...
        }
        function main() \{
          command1();
          command2();
          foo2($bar);
          ...
        }
```

All the commands in one function is combined with the curly brackets. Function ends to return() or exit() command, or to the outmost brancet (returns 0).

You can include another source code file to the program in any place with the command #include filename.

A comment in a source file is text that does not affect what the program does, and is meant purely for explaining or describing something. In a BZ source file, whenever the symbol // is encountered on a line, all the characters from that point on until the end of the line are considered to be a comment and are ignored. Also everything between /* and */ is considered as comment.

1.4 Constants

A string constant is a sequence of ASCII characters enclosed in quotes, for example, "Hello", "World", or "Hello World". Some character can, or must be, inserted by quoting it with '\' character.

Quote	Character	ASCII
\"	"	34
\\	\	92
\a	Bell	7
\b	Backspace	8
\e	Escape	27
\f	Form Feed	12
\n	Line Feed	10
\r	Carriage Return	13
\t	Horizontal Tabulator	9
\v	Vertical Tabulator	11

All other characters can be quoted by writing a backslash followed by character's ASCII number in octal (base-8). For example ESC (\e) can also be written "\033".

An integer constant is a sequence of digits representing an integer

value in the range -2147483648 to 2147483647. An integer constant must start with a digit from 0 to 9 or the negative sign '-' followed by a digit.

1.5 Funcs and Vars

You can callunctions by writing it's name just like it would be an internal BZ command. You must define function before you can call it!

```
Example  foobar();
```

You can also use variables. A variable is a location in memory where something is stored. The contents of a variable can be changed by program code (hence the name). In BZ, there are two types of variables, integer variables, and string variables. The former holds an integer value while the latter holds a text string. Depending on where it is defined, a variable is either global or local. If a variable is global, it means that it can be used by any part of the script after the point where it is defined. If a variable is local, it means that it can only be used by the part of the script to which it is local, for example, the function inside which it is defined.

A variable name can be up to 16 digits long, and may include the letters 'A' to 'Z', 'a' to 'z', the digits '0' to '9', or the underscore character '_'. The name must start with '$' character.

```
Example  $foo=43;
         $bar="foobar!";
```

Every variable (except pre-defined (except $bv_txt)) is an index variable. You can refer to variable just by it's name (which actually is index 0), or by using name and number in brackets.

```
Example  $foo="index 0";
         $foo[0]="index 0";
         $foo[1]="index 1 in $foo";
         $foo[300]="index 300 in $foo";
```

Index number can be any number (or expression, of course) from 0 to 65535. If you need multi-dimensional variables, you can do it by yourself! Let's say you need 10 * 10 * 10 index variable (3-dimensional). You just refer to it as $name[((x*10)+y)*10)+z], where x, y and z are indexes.

You can not pass index variables from function to another, but you can pass one value of it.

```
Example  function foo($bar) {
           ...
         }
         function main() {
           var $index;
           ...
           foo($index[4]);
           foo($index);          // Remember, this is index 0!
```

```
    ...
  }
```

You can combine variables and functions and do different calculations with them.

```
Example  $buz=3*($foo+3)-2;
         $fub=$buz*foobar();
```

The following table lists the operators available in BZ:

Symbol	(Un/Bin)ary	What it is/does
-	unary	Arithmetic negation
!	unary	Logical NOT
*	binary	Multiplication
/	binary	Division
%	binary	Remainder (Mod)
+	binary	Addition
-	binary	Subtraction
<	binary	Less than
>	binary	Grater than
<=	binary	Less than or equal to
>=	binary	Greater than or equal to
==	binary	Equality
!=	binary	Inequality
&	binary	Bitwise AND
|	binary	Bitwise OR
^	binary	Bitwise Exclusive OR
&&	binary	Logical AND
||	binary	Logical OR
=	binary	Assignment
<<	binary	Shift left
>>	binary	Shift right

You can also open up to 10 different files from your program. These filehandles are numbered from 0 to 9. Remember to close all the files you open!

1.6 The BZ run-time library

commands:

asm

bbbs

beep

break

copy

delay

```
delete

do..while

exit

fclose

feof

fgets

fopen

for

fprintf

fseek

ftell

getch

getenv

getnodestatus

help

if

input

kbhit

make_pcboard12sys

make_pcboard14sys

parsecom

pos

printf

regexp

remove

rename

return

seekforuser
```

sendnode

setenv

showfile

spawn

sprintf

stlocase

strcat

strlen

stupcase

switch

system

tictactoe

var

while

yellsysop

1.7 asm

Function Command is used to program BZ49 processor directly. Real Guru uses only this one command.

Syntax asm(command,value);

Return Value None.

Example asm(label,49);
asm(chop,0);

1.8 bbbs

Function Command executes BBBS-command.

Syntax bbbs(command);

Return Value None.

See Also

```
               system
                      Example  bbbs("f n,,f get z g y");
```

1.9 beep

```
     Function  Command makes one beep-sound to the user.

       Syntax  beep();

 Return Value  None.

      Example  beep();
```

1.10 break

```
                     Function  Jumps out of while, do..while, for or switch loop.

       Syntax  break;

 Return Value  None.

     See Also
               while
               ,
               do..while
               ,
               for
               ,
               switch
                      Example  for ($bar=0;;$bar=$bar+1) if ($bar==3) break;
```

1.11 copy

```
                     Function  Copies part of string.

       Syntax  copy(from,start,length);

 Return Value  String.

     See Also
               delete
                      Example  $bar=copy("123456",3,2);
```

1.12 delay

```
     Function  Waits specified time time in 1/10 seconds.

       Syntax  delay(time);

 Return Value  None.

      Example  delay(30);
```

1.13 delete

```
                    Function  Deletes part of string.

       Syntax  delete(from,start,length);

 Return Value  String.

     See Also
               copy
                      Example  $bar=delete("123456",3,2);
```

1.14 do..while

```
                    Function  Repeat command while condition is true.

       Syntax  do command while (condition);

      Remarks  Command  will be executed at least once and will continue to be
               executed repeatedly until the condition becomes false.

 Return Value  None.

     See Also
               break
               ,
               while
                      Example  do {
                 $bar=input("Command: ",9,1);
               } while ($bar!="quit");
```

1.15 exit

```
                    Function  Exits your  program  immediately.  Program  is ↩
                       terminated with
               errorlevel code. All the files must be closed before this!

       Syntax  exit(code);

 Return Value  Never returns.
```

```
    See Also
              return
              ,
              fclose
                     Example  exit(49);
```

1.16 fclose

```
                    Function  Closes  the  file filehandle opened with fopen.  ↩
                       You must always
              close the files you open!

      Syntax  fclose(filehandle);

Return Value  None.

    See Also
              fopen
                     Example  fclose(4);
```

1.17 feof

```
                    Function  Determines  whether  the  end of  the file  ↩
                       filehandle has been
              reached. It can be checked after reading the file.

      Syntax  feof(filehandle);

Return Value  True, if end-of-file has been reached.

    See Also
              ftell
                     Example  while ($bv_carrier) {
                $s=fgets(0);
                if (feof(0)) break;
                printf("%s\n",$s);
              }
```

1.18 fgets

```
                    Function  Reads one line from the file filehandle.

      Syntax  fgets(filehandle);

Return Value  String.

    See Also
              fprintf
                     Example  $bar=fgets(9);
```

1.19 fopen

```
      Function  Opens the  file filename for input or output,  ↩
         using filehandle
number filehandle.

  Syntax  fopen(filehandle,filename,mode);

          Modes are:

          "rt"    read text
          "wt"    write text
          "at"    append text
          "rb"    read binary
          "wb"    write binary
          "ab"    append binary

Return Value  True, if file opened.

   See Also
          fclose
           Example  fopen(6,"output.fil","wt");
```

1.20 for

```
      Function  Command  is used to loop continuously while a  ↩
         certain condition
is true.

  Syntax  for ({expression{,...}}; {condition{,...}};
            {expression{,...}) statement;

   Remarks  The  first  expression  is  the  one that should initialize the
          count  variable.  The second parameter is the conditional test.
          As  long  as  it  evaluates  to  true, the  statement  will be
          executed.  The third  expression  is the  one that is used to
          increment the count variable.

          Expression can be empty or one or more expressions separated by
          commas. Condition can be empty (true) or one or more conditions
          separated by commas (logical and).

Return Value  None.

   See Also
          do..while
           Example  for ($bar=0; $bar<4; $bar=$bar+1) {
            $foo=$foo+$bar;
            printf("$bar = %u\n",$bar);
          }
          for ($bar=0, $foo=3; $foo<9 && $buz; )
            $bar=$foo=$foo+1;
          for (;;) printf("Forever!\n");
```

1.21 fprintf

```
      Function  Writes formatted text to the file filehandle.

Syntax  fprintf(filehandle,format,argument{,argument...});

Return Value  None.

    See Also
            printf
             Example  fprintf(5,"Foobar");
            fprintf(7,"Bar=%s, 3+3=%u\n",$bar,3+3);
```

1.22 fseek

```
      Function  Sets  the   file pointer to a new position that is ↩
          offset bytes
            from  the  file  location given by whence. For text mode files,
            offset should be 0 or a value returned by ftell.

Syntax  fseek(filehandle,offset,whence);

 Remarks  Whence must be one of the values 0, 1 or 2, as follows:

            whence   C-define        File location
            ==================================================
            0        SEEK_SET        File beginning
            1        SEEK_CUR        Current file pointer position
            2        SEEK_END        End-of-file

Return Value  True, if error encounted.

    See Also
            ftell
             Example  fseek(0,0,0);
            fseek(5,-5,2);
            if (fseek(1,0,1))
              printf("Error: File #1 is closed!");
```

1.23 ftell

```
      Function  Returns  the  current file  pointer. The offset is ↩
          measured in
            bytes  from  the beginning of the file (if the file is binary).
            The value returned by ftell can be used in a subsequent call to
            fseek.

Syntax  ftell(filehandle);

Return Value  Number.

    See Also
```

```
              fseek
                     Example  fseek(4,0,2);
              printf("Filelength is %u bytes.",ftell(6));
```

1.24 getch

```
                    Function  Reads one keypress from the user.

       Syntax  getch();

  Return Value  Character.

      See Also
              input
                     Example  printf("You pressed \"%c\" key.\n",getch());
```

1.25 getenv

```
                    Function  Reads environtment variable.

       Syntax  getenv(envname);

  Return Value  String.

      See Also
              setenv
                     Example  printf("%s\n",getenv("WHOTEXT));
```

1.26 getnodestatus

```
      Function  Reads node's status and returns one item of it.

       Syntax  getnodenumber(node,item);

       Remarks  Available items are:

                        Item     Meaning         Return Value
                        ===================================
                        0        split           number
                        1        bstatus         number
                        2        zstatus         number
                        3        speed           number
                        4        time            number
                        5        endtime         number
                        6        name            string
                        7        realname        string

                        See bbbsdef.h for more information.

  Return Value  String or number.
```

```
Example  printf("User in node 1 is %s.\n",getnodestatus(1,7));
```

1.27 help

```
Function  Executes AmigaGuide reader for given node in file filename.
          If filename is an empty string, standard helpfile is assumed.

  Syntax  help(filename,nodename);

Return Value  None.

Example  help("","Main");
         help("/usr/local/etc/foohelp","foo");
```

1.28 if

```
Function  Command is used when a command or group of commands should be
processed only if a condition is true.

  Syntax  if (condition) command1;
          else command2;

Return Value  None.

See Also
          switch
          Example  if ($buz=49) printf("Yeah!");
          if ($bar==49) {
            $buz=30;
            printf("It was true!");
          } else {
            $buz=0;
            printf("False.");
          }
```

1.29 input

```
Function  Reads user's input to your program. Maximum length characters
are read. If multi is true then only the first word is read.

  Syntax  input(prompt,length,multi);

Return Value  String.

See Also
          getch
          Example  $command=input("Your Command: ",50,1);
```

```
$foo=input("Enter string: ",50,0);
```

1.30 kbhit

```
Function  Checks whether user has pressed any key or not.

  Syntax  kbhit();

Return Value  True, if key is pressed.

  See Also
          getch
           Example  while (kbhit()) getch();
```

1.31 make_pcboard12sys

```
  Function  Makes PCBOARD.SYS-style v12 control file.

    Syntax  make_pcboard12sys(filename);

Return Value  None.

   Example  make_pcboard12sys("pcboard.sys");
```

1.32 make_pcboard14sys

```
  Function  Makes PCBOARD.SYS-style v14 control file.

    Syntax  make_pcboard14sys(filename);

Return Value  None.

   Example  make_pcboard14sys("pcboard.sys");
```

1.33 parsecom

```
  Function  Parses user's input.

    Syntax  parsecom(commands,input);

Return Value  0        Empty input string
              1        First command
              2        Second command
              ...
              255      Unknown command

   Example  parsecom("Quit/Read/UGH",input("Com: ",40,1));
```

1.34 pos

```
                Function  Seeks for string1 from string2. If found, returns ↩
                   the startting
             character number. If not, returns zero.

       Syntax  pos(string1,string2);

 Return Value  Number.

     See Also
             regexp
                    Example  $bar=pos($foo,"ABCDEF");
```

1.35 printf

```
                Function  Writes formatted text to the screen.

       Syntax  printf(format{,argument{,argument...}});

      Remarks  A format specification, which consists of optional and required
             fields, has following form: %{flags}{width}type

             Available types are:

            type     Output
             ========================================================
             %        The character '%'
             c        Single character
             d        Signed decimal integer
             i        Signed decimal integer
             o        Unsigned octal integer
             s        String
             u        Unsigned decimal integer
             x        Unsigned hexadecimal integer, using "abcdef"
             X        Unsigned hexadecimal integer, using "ABCDEF"

             Available flags are:

             Flag     Meaning
             ============================================================
             -        Left justify the result within the given field
                      width
             +        Prefix the output value with a sign
             0        Zeros are added until the minumum width is
                      reached

 Return Value  None.

     See Also
             fprintf
             ,
             sprintf
                    Example  printf("Foobar!\n");
```

```
printf("Foobar, %s!\n",$bu_name);
printf("49=0x%04X=%4oo",49,49);
printf("%-33s%s",$bu_name,$bu_city);
```

1.36 regexp

```
Function  Executes regular expression matching for given strings.

  Syntax  regexp(match,string);

Return Value  True, if match can be found from string.

  See Also
pos
   Example  if (regexp($input,$r=fgets())) printf("%s\n",$r);
```

1.37 remove

```
Function  Deletes the file filename. Wildcards are not allowed here. If
you want to delete multiple files, use system command.

  Syntax  remove(filename);

Return Value  None.

  See Also
system
   Example  remove("c:/bbs/main/tempfile");
```

1.38 rename

```
  Function  Renames the file to specific name/dir.

  Syntax  rename(oldfile,newfile);

Return Value  None.

  Example  rename("c:/bbs/curuser","c:/bbs/lastuser");
```

1.39 return

```
Function  Returns from function to it's caller. The number is the value
that should be returned. In main program return and exit both
exits the program.
```

```
       Syntax  return(value);

 Return Value  Never returns.

     See Also
               exit
                      Example  return(3+$bar);
```

1.40 seekforuser

```
     Function  Returns user's name usernumber.

       Syntax  seekforuser(name);

 Return Value  65535 if not found, <65535 if found

      Example  seekforuser("Demo Luser");
```

1.41 sendnode

```
     Function  Sends a  nodemessage message to node to from node from with a
               nodemessage filter level filter.

       Syntax  sendnode(from,to,message,filter);

      Remarks  BBBS has some "control"-nodemessages:

               Message           Action when received
               ======================================================
               \02CLICK         reply with "\02HUM"
               \02UDATA         re-read user record
               \02RGROUP        re-read groups file

               (\02 = ASCII character 2, not number "02")

 Return Value  None.

      Example  sendnode(42,$bv_nodenumber,"Hello!",255);
```

1.42 setenv

```
                    Function  Makes an environment variable.

       Syntax  setenv(envname,text);

 Return Value  None.

     See Also
               getenv
```

```
       Example  setenv("WHOTEXT",$bu_name);
setenv("FOO","");
```

1.43 showfile

```
     Function  Shows file filename to the screen.

       Syntax  showfile(filename);

Return Value  None.

      Example  showfile("c:\\bbs\\menus\\bull0");
```

1.44 spawn

```
     Function  Executes another BZB-file.

       Syntax  spawn(filename);

Return Value  None.

      Example  spawn("c:/bbs/jukka");
```

1.45 sprintf

```
            Function  Writes formatted text to a string.

       Syntax  sprintf(format,argument{,argument...});

Return Value  String.

     See Also
               printf
       Example  $bar=sprintf("49=0x%04X=%oo",49,49);
```

1.46 stlocase

```
            Function  Converts string to lower case.

       Syntax  stlocase(string);

Return Value  String.

     See Also
               stupcase
       Example  $bar=stlocase($bar);
```

1.47 strcat

```
     Function  Joins two strings to one.

       Syntax  strcat(string1,string2);

 Return Value  String.

      Example  printf(strcat("Hello, ",
                 input("Your name: ",20)));
```

1.48 strlen

```
     Function  Returns the length of string.

       Syntax  strlen(string);

 Return Value  Number.

      Example  $length=strlen("123456");
```

1.49 stupcase

```
                    Function  Converts string to upper case.

       Syntax  stupcase(string);

 Return Value  String.

     See Also
               stlocase
                      Example  $bar=stupcase($bar);
```

1.50 switch

```
                     Function  switch,  case  and  default  commands  are used  ↩
                        to create big
               if-else loops with same kind of conditions.

       Syntax  switch (expression1) \{
                 {case expression2:
                 {case expression2:{...}}{statement}}
                 {...}
                 {default:{statement}}
               \};

 Return Value  None.

     See Also
```

```
if
       Example  switch ($bar) {
  case 1:
  case 2:{
           printf("1 or 2\n");
           break;
         }
  case 3:{
           printf("3\n");
           break;
         }
  case 4:printf("4 or ");
  case 5:{
           printf("5\n");
           break;
         }
  default:{
           printf("unknown\n");
           break;
         }
}
```

1.51 system

```
                Function  Executes operating system command.

       Syntax  system(command,swap);

 Return Value  Number, errorlevel returned by the operating system.

     See Also
               bbbs
                Example  printf("Errorlevel=%u",system("foo.exe",1));
```

1.52 tictactoe

```
     Function  Executes TicTacToe-game.

       Syntax  tictactoe(mode,mark);

 Return Value  None.

      Example  tictactoe(1,0);
               tictactoe(2,1);
               tictactoe(2,2);
```

1.53 var

```
      Function  Defines variables.

        Syntax  var variable {, variable};

  Return Value  None.

       Example  var $bar;
                var $foo, $buz, $fuu;
```

1.54 while

```
                      Function  Command  is used to loop continuously while a  ↩
                         certain condition
                is true.

        Syntax  while (condition) commands;

       Remarks  Command  would  continue  to  be  repeated  over and over while
                condition is true. Note that if the condition is false from the
                beginning, command will never be executed.

  Return Value  None.

      See Also
                do..while
                       Example  while ($bar<5) $foo=$foo+($bar=$bar+1);
```

1.55 yellsysop

```
      Function  Pages SysOp.

        Syntax  yellsysop(ask);

       Remarks  If ask is nonzero, yellsysop asks a reason for chat.

  Return Value  True, if SysOp answered.

       Example  yellsysop(0);
```

1.56 BBBS Variables

```
                Additionally  there  are  following  variables already  defined  ↩
                   in  BZ
language:


                 bc - Conference data

                 bf - Functions
```

```
bg - Global config

bl - Local config

bn - Node data

bu - User data

bv - Internal variables
       -------------------------

List of all variables

Bitwise operation
```

1.57 Conference data

```
          Conference data is variables that are related to the conferences.

SUBTOPICS:

          $bc_count_of_co

          $bc_post_conf

          $bc_resume_conf

          $bc_fileinfo_co

          $bc_lastread

          $bc_status

          $bc_confname

          $bc_description
```

1.58 $bc_count_of_co

```
   Variable   $bc_count_of_co

       Type   number (constant)

Description   Returns the total number of conferences.

              Limit is 2048 for DOS version and 32000 for OS/2 version.
```

1.59 $bc_post_conf

```
   Variable   $bc_post_conf

       Type   number (constant)

Description   Conference number for the defined post conference.
              If no post conference is defined this will contain the
              number 65535.
```

1.60 $bc_resume_conf

```
   Variable   $bc_resume_conf

       Type   number (constant)

Description   Conference number for the defined resume conference.
              If no resume conference is defined this will contain the
              number 65535.
```

1.61 $bc_fileinfo_co

```
   Variable   $bc_fileinfo_co

       Type   number (constant)

Description   Conference number for the defined fileinfo conference.
              If no fileinfo conference is defined this will contain the
              number 65535.
```

1.62 $bc_lastread

```
   Variable   $bc_lastread

       Type   number (constant)

Description   Returns the message number that the user last read.
              Will not change until the user logs out normally.
```

1.63 $bc_status

```
                   Variable   $bc_status

       Type   number (constant)

Description   Conference status Returns a number which indicates
              what status the current conference has.

              The bits are:
```

```
             1     must
             2     member
             4     invite
             8     fidoarea
             16    postarea
             32    allowpriv
             64    nomarks
             128   noreply
             256   nostrip
             512   allfix
             1024  namefix
             2048  alias
             4096  sempty5
             8192  sempty6
             16384 moderated      /* used internally, DO NOT ALTER! */
             32768 sempty7

   See also
               Bitwise operation
```

1.64 $bc_confname

```
   Variable   $bc_confname

       Type   string (constant)

Description   Returns the name of the current conference.
```

1.65 $bc_description

```
   Variable   $bc_description

       Type   string (constant)

Description   Returns the description of the current conference.
```

1.66 Functions

```
                       SUBTOPICS:

                  $bf_timleft

                  $bf_hour

                  $bf_min

                  $bf_sec

                  $bf_year
```

```
$bf_month

$bf_day
```

1.67 $bf_timleft

```
   Variable   $bf_timleft

       Type   number (constant)

Description   Returns minutes left for this call for the current user.
```

1.68 $bf_hour

```
   Variable   $bf_hour

       Type   number (constant)

Description   This is the hour part of the time. It returns the
              current hour.
```

1.69 $bf_min

```
   Variable   $bf_min

       Type   number (constant)

Description   This is the minute part of the time. It returns the
              current minutes.
```

1.70 $bf_sec

```
   Variable   $bf_sec

       Type   number (constant)

Description   This is the seconds part of the time. It returns the
              current seconds.
```

1.71 $bf_year

```
   Variable   $bf_year

       Type   number (constant)
```

```
Description   This is year part of the date. It returns the
              current year.
```

1.72 $bf_month

```
   Variable   $bf_month

       Type   number (constant)

Description   This is month part of the date. It returns the
              current month.
```

1.73 $bf_day

```
   Variable   $bf_day

       Type   number (constant)

Description   This is the day part of the date. It returns the
              current day.
```

1.74 Global config

```
                      SUBTOPICS:

                $bg_bbbs_name

                $bg_sysop_name

                $bg_closed_pass

                $bg_grabfile

                $bg_maindir

                $bg_updir

                $bg_tempdir

                $bg_menudir

                $bg_tickdir

                $bg_netmail

                $bg_inbound

                $bg_max_nodes

                $bg_bankmax
```

```
$bg_newu_time

$bg_bankrate

$bg_msgrate

$bg_newu_access

$bg_gtoggles
```

1.75 $bg_bbbs_name

```
   Variable   $bg_bbbs_name

       Type   string (constant)

Description   Returns the name of the board.
```

1.76 $bg_sysop_name

```
   Variable   $bg_sysop_name

       Type   string (constant)

Description   Returns the name of User #0, which is the "real" SysOp.
```

1.77 $bg_closed_pass

```
   Variable   $bg_closed_pass

       Type   string (constant)

Description   Returns the password for accessing the system if you
              run a closed system. This string is crypted.
```

1.78 $bg_grabfile

```
   Variable   $bg_grabfile

       Type   string (constant)

Description   Returns the name of the grabfile as you have defined
              it in BCFG/4.
```

1.79 $bg_maindir

```
   Variable   $bg_maindir

       Type   string (constant)

Description   Returns the path to your main dir including a
              trailing slash.
```

1.80 $bg_updir

```
   Variable   $bg_updir

       Type   string (constant)

Description   Returns the path to your upload directory including
              a trailing slash.
```

1.81 $bg_tempdir

```
   Variable   $bg_tempdir

       Type   string (constant)

Description   Returns the path to your temp directory including
              a trailing slash.
```

1.82 $bg_menudir

```
   Variable   $bg_menudir

       Type   string (constant)

Description   Returns the path to your main menu directory
              including a trailing slash.
```

1.83 $bg_tickdir

```
   Variable   $bg_tickdir

       Type   string (constant)

Description   Returns the path to your tick files directory
              including a trailing slash.
```

1.84 $bg_netmail

```
   Variable   $bg_netmail

       Type   string (constant)

Description   Returns the path to your netmail directory
              including a trailing slash.
```

1.85 $bg_inbound

```
   Variable   $bg_inbound

       Type   string (constant)

Description   Returns the path to your inbound files directory
              including a trailing slash.
```

1.86 $bg_max_nodes

```
   Variable   $bg_max_nodes

       Type   number (constant)

Description   Returns the number of nodes configured in your system.
```

1.87 $bg_bankmax

```
   Variable   $bg_bankmax

       Type   number (constant)

Description   Returns the maximum amount of time allowed to store
              in the Time Bank.
```

1.88 $bg_newu_time

```
   Variable   $bg_newu_time

       Type   number (constant)

Description   Returns the default timelimit for new users.
```

1.89 $bg_bankrate

```
   Variable   $bg_bankrate

       Type   number (constant)

Description   Returns the ratio divider used for calculating the
              Time Bank ratio.
```

1.90 $bg_msgrate

```
   Variable   $bg_msgrate

       Type   number (constant)

Description   Returns the ratio divider used for calculating the
              number of messages a user must write to get Download access
              from BRoboCop.
```

1.91 $bg_newu_access

```
                   Variable   $bg_newu_access

       Type   number (constant)

Description   Returns a number which sets several different user
              settings:

              0x40000000        New user has download access
              0x80000000        New user has upload access

   See also
                Bitwise Operation
```

1.92 $bg_gtoggles

```
                   Variable   $bg_gtoggles

      Type    number (constant)

Description   Returns a number containg different global toggles.

              Bit Meaning
              === ========================
               0  up_down_check
               1  show_privates
               2  brobocop
               3  hippo
               4  show_empty
```

```
 5  whotext
 6  pack_messages
 7  fido_hydra
 8  fido_zedzap
 9  fido_tranx
10  fido_unlistnode
11  fido_unlistpoint
12  fido_unprotnode
13  fido_freq_answering
14  fido_freq_calling
15  show_sysop_in_stats
16  bmt_check_destination
17  disable_newu_address
18  disable_newu_birthday
19  users_are_hidden
20  upload_scan
21  poll_all_crashes
22  delete_dup_uploads
23  savebad
24  savesecure
25  bmt_log_headers
27  no_remote_sysop

See also
             Bitwise Operation
```

1.93 Local config

```
                    SUBTOPICS:

               $bl_modem_init1

               $bl_modem_init2

               $bl_modem_init3

               $bl_modem_hangu

               $bl_modem_busy_

               $bl_modem_answe

               $bl_fd_dobbs

               $bl_logfile

               $bl_loginlog

               $bl_grabdir

               $bl_menudir

               $bl_newdir

               $bl_start_speed
```

```
$bl_min_speed

$bl_base_addres

$bl_irq

$bl_pollrate

$bl_answer_ring

$bl_ltoggles
```

1.94 $bl_modem_init1

```
   Variable   $bl_modem_init1

       Type   string (constant)

Description   Returns the first of the three line Init string from BCFG/4.
```

1.95 $bl_modem_init2

```
   Variable   $bl_modem_init2

       Type   string (constant)

Description   Returns the second of the three line Init string from BCFG/4.
```

1.96 $bl_modem_init3

```
   Variable   $bl_modem_init3

       Type   string (constant)

Description   Returns the third of the three line Init string from BCFG/4.
```

1.97 $bl_modem_hangu

```
   Variable   $bl_modem_hangu

       Type   string (constant)

Description   Returns the configured string to force the modem to hang up.
```

1.98 $bl_modem_busy_

```
   Variable   $bl_modem_busy_

       Type   string (constant)

Description   Returns the configured string to set the modem busy from BCFG/4.
```

1.99 $bl_modem_answe

```
   Variable   $bl_modem_answe

       Type   string (constant)

Description   Returns the configured string to tell the modem to answer an
              incoming call. This is configured in BCFG/4.
```

1.100 $bl_fd_dobbs

```
   Variable   $bl_fd_dobbs

       Type   string (constant)

Description   Returns the path and name of the DOBBS.BAT file used for
              FrontDoor if one is defined in BCFG/4. If none is defined, the
              string will be empty.
```

1.101 $bl_logfile

```
   Variable   $bl_logfile

       Type   string (constant)

Description   Returns the path and the filename of the current
              node's log file.
```

1.102 $bl_loginlog

```
   Variable   $bl_loginlog

       Type   string (constant)

Description   Returns the path and the filename of the current node's
              login file.
```

1.103 $bl_grabdir

```
   Variable   $bl_grabdir

       Type   string (constant)

Description   Returns the path including a trailing slash for the
              current node's grab directory.
```

1.104 $bl_menudir

```
   Variable   $bl_menudir

       Type   string (constant)

Description   Returns the path including a trailing slash for the
              current node's menu directory.
```

1.105 $bl_newdir

```
   Variable   $bl_newdir

       Type   string (constant)

Description   Returns the path including a trailing slash for the
              current node's temporarly upload directory. (The directory that
              BBBS uses to store files users upload while the upload is in
              process.)
```

1.106 $bl_start_speed

```
   Variable   $bl_start_speed

       Type   number (constant)

Description   Returns the configured start speed from BCFG/4.
```

1.107 $bl_min_speed

```
   Variable   $bl_min_speed

       Type   number (constant)

Description   Returns the configured minimum speed to log in from BCFG/4.
```

1.108 $bl_base_addres

```
   Variable   $bl_base_addres

       Type   number (constant)

Description   Returns the configured base I/O address for the
              communications port for the current node.
```

1.109 $bl_irq

```
   Variable   $bl_irq

       Type   number (constant)

Description   Returns the configured IRQ address for the communications
              port for the current node.
```

1.110 $bl_pollrate

```
   Variable   $bl_pollrate

       Type   number (constant)

Description   Returns the configured polling rate for current node.
```

1.111 $bl_answer_ring

```
   Variable   $bl_answer_ring

       Type   number (constant)

Description   Returns the configured number of rings to answer on for
              the current node.
```

1.112 $bl_ltoggles

```
                   Variable   $bl_ltoggles

       Type   number (constant)

Description   Returns a number containing several different settings
              that are unique to each node. The numbers are:

              Bit Meaning
              === ====================
               0  local_bell
```

```
 1  rts_cts
 2  reset_speed
 3  hangup_at_exit
 4  set_16550
 5  local_sysop_keys
 6  local_echo
 7  save_screen
 8  null_modem_login
 9  send_crashmail
10  backdoor
11  fast_fax
12  dont_check_carrier
13  nocarrier_is_busy
14  show_shell_output
15  allow_shell_break
```

See also

Bitwise Operation

1.113 Node data

SUBTOPICS:

$bn_split

$bn_bstatus

$bn_zstatus

$bn_speed

$bn_time

$bn_name

$bn_realname

1.114 $bn_split

Variable $bn_split

Type number

Description Currently unused, should be zero.

1.115 $bn_bstatus

Variable $bn_bstatus

Type number

```
Description   If bit 0 is set, user has error correcting modem.
              If bit 1 is set, this is a mail session.
```

1.116 $bn_zstatus

```
   Variable   $bn_zstatus

       Type   number

Description   Enumed activity:

               0 = logged off
               1 = active
               2 = not active
               3 = writing
               4 = grab
               5 = down
               6 = up
               7 = chat
               8 = door
               9 = groupchat
              10 = telnet
```

1.117 $bn_speed

```
   Variable   $bn_speed

       Type   number

Description   Line connect speed, 0 if local.
```

1.118 $bn_time

```
   Variable   $bn_time

       Type   number

Description   Login time, hours*256+minutes.
```

1.119 $bn_name

```
   Variable   $bn_name

       Type   string

Description   Returns the name used in the Who command. This can be
              set by the user using U SET "WHOTEXT"
```

1.120 $bn_realname

```
   Variable   $bn_realname

       Type   string

Description   Returns the realname of current user.
```

1.121 User data

```
                User data is variables that are related to the user.

        SUBTOPICS:

                 $bu_name

                 $bu_address

                 $bu_city

                 $bu_phone

                 $bu_birth

                 $bu_ok2login

                 $bu_termcap

                 $bu_pagelength

                 $bu_charset

                 $bu_language

                 $bu_readmode

                 $bu_packtype

                 $bu_protocol

                 $bu_nodemsgfilt

                 $bu_timelimit

                 $bu_timeleft

                 $bu_timeson

                 $bu_msgleft

                 $bu_uploaded

                 $bu_downloaded
```

```
$bu_pmsgleft
$bu_pfileup
$bu_pfiledown
$bu_pcalled
$bu_fchecked
$bu_timebank
$bu_resume
$bu_limits
$bu_access
$bu_utoggles
$bu_lasttime
$bu_msgread
$bu_msgdumped
$bu_bytesup
$bu_bytesdown
$bu_userbits
```

1.122 $bu_name

```
   Variable   $bu_name
       Type   string (constant)
Description   Returns the users name as written to BBBS in uppercase.
Example code  printf("%s\n",$bu_name);
```

1.123 $bu_address

```
   Variable   $bu_address
       Type   string
Description   Returns the users address as written to BBBS.
Example code  printf("%s\n",$bu_address);
```

1.124 $bu_city

```
   Variable   $bu_city

       Type   string

Description   Returns the users city as written to BBBS.

Example code  printf("%s\n",$bu_city);
```

1.125 $bu_phone

```
   Variable   $bu_phone

       Type   string

Description   Returns the users phonenumber as written to BBBS.

Example code  printf("%s\n",$bu_phone);
```

1.126 $bu_birth

```
   Variable   $bu_birth

       Type   string

Description   Returns the users birthday as written to BBBS.

Example code  printf("%s\n",$bu_birth);
```

1.127 $bu_ok2login

```
   Variable   $bu_ok2login

       Type   number

Description   Contains users login status. There are three different
              types:

              0 = yes      The user can log in.
              1 = getlost  The user is shown the getlost file and then
                           logged off.
              2 = killed   The user name is removed, but a new user with
                           that name can log in as a new user.

Example code  printf("Login status: %s\n",$bu_ok2login);
```

1.128 $bu_termcap

```
                  Variable   $bu_termcap

       Type   number

Description   This variable contains terminal related information.

              It is divided into two bitwise parts:

              Bit 0-3 - terminal types:
              0=TTY
              1=ANSI
              2=AVATAR
              3=VT320

              Bit 4-7 - editors:
              0=Line
              1=FSE
              2=MG

   See also
                Bitwise operation
```

1.129 $bu_pagelength

```
   Variable   $bu_pagelength

       Type   number

Description   Sets number of lines pr page for the current user.

Example code  function main() {
                var $p;
                $p=$bu_pagelength;
                $bu_pagelength=0;
                ..do your stuff that requires no --more-- prompt..
                $bu_pagelength=$p;
              }
```

1.130 $bu_charset

```
   Variable   $bu_charset

       Type   number

Description   Returns a number representing the character set the
              user is using. The number for the different character sets are:

               0 IBM
               1 SF7
               2 ISO
```

```
 3 IBN
 4 US7
 5 GE7
 6 NO7
 7 FR7
 8 IT7
 9 SP7
10 MAC
```

1.131 $bu_language

```
   Variable   $bu_language

       Type   number

Description   Returns the users language. Values are 0  through 9.
             (0=English,1=Suomi,2=Svenska,3=Norsk)
```

1.132 $bu_readmode

```
                  Variable   $bu_readmode

       Type   number

Description   This contains information about read modes for the
              user and preferred format for grab collection. It is divieded
              into two bitwise parts:

              Bit 0-3 - read modes:
              0=Marked
              1=Reference
              2=Forward

              Bit 4-7 - offline format:
              0=Text
              1=Hippo
              2=Hippo2
              3=OMEN
              4=QWK

   See also
               Bitwise operation
```

1.133 $bu_packtype

```
   Variable   $bu_packtype

       Type   number

Description   Contains a number corresponding to a packer type.
```

```
              The different types are:

              0=text
              1=arc
              2=zip
              3=lzh
              4=arj
              5=zoo
              6=hpk

              Additional packers will just be given the next number in range.
```

1.134 $bu_protocol

```
   Variable   $bu_protocol

       Type   number

Description   Returns a number corresponding to a protocol.

              The different types are:

              0=Zmodem       (can be internal)
              1=Ymodem       (can be internal)
              2=Xmodem       (can be internal)
              3=BiModem
              4=Xmodem CRC   (can be internal)
              5=Ymodem Batch (can be internal)
              6=HSLink
              7=Hydra        (can be internal)
              8=ZedZap       (can be internal)

              Additional protocls will just be given the next number in range.
```

1.135 $bu_nodemsgfilt

```
   Variable   $bu_nodemsgfilt

       Type   number

Description   Returns the node message filter level.

              Different levels are:
               0 nothing
              10 login / logout
              20 entered message
              30 joined/exited group chat
              49 messages in group chat, node messages
              50 everything
```

1.136 $bu_timelimit

```
   Variable   $bu_timelimit

       Type   number

Description   Returns the timelimit for the current call.
```

1.137 $bu_timeleft

```
   Variable   $bu_timeleft

       Type   number

Description   Returns the number of minutes left for the current call.
```

1.138 $bu_timeson

```
   Variable   $bu_timeson

       Type   number

Description   Returns the number of logins for all time for the
              current user.
```

1.139 $bu_msgleft

```
   Variable   $bu_msgleft

       Type   number

Description   Returns the number of messages written for the current
              user for all time.
```

1.140 $bu_uploaded

```
   Variable   $bu_uploaded

       Type   number

Description   Returns the number of files uploaded by the current user.
```

1.141 $bu_downloaded

```
   Variable   $bu_downloaded

       Type   number

Description   Returns the number of files dowloaded by the current user.
```

1.142 $bu_pmsgleft

```
   Variable   $bu_pmsgleft

       Type   number

Description   Returns the number of messages left since the last reset
              for the current user.
```

1.143 $bu_pfileup

```
   Variable   $bu_pfileup

       Type   number

Description   Returns the number of files upload since the last reset
              for the current user.
```

1.144 $bu_pfiledown

```
   Variable   $bu_pfiledown

       Type   number

Description   Returns the number of files downloaded since the last
              reset for the current user.
```

1.145 $bu_pcalled

```
   Variable   $bu_pcalled

       Type   number

Description   Returns the number of calls since the last reset for the
              current user.
```

1.146 $bu_fchecked

```
                  Variable   $bu_fchecked

       Type   number

Description   Returns a datecode for the date the current user last
              checked for new files.

              Day is lower 5 bits, month is next 4 bits and year (since 1980)
              is the upper 7 bits.

   See also
                Bitwise operation
```

1.147 $bu_timebank

```
   Variable   $bu_timebank

       Type   number

Description   Returns the number of minutes the current user has
              stored in his/hers timebank.
```

1.148 $bu_resume

```
   Variable   $bu_resume

       Type   number

Description   Returns the message number in the User Info conference
              that contains the current users userresume.
```

1.149 $bu_limits

```
   Variable   $bu_limits

       Type   number

Description   Returns a number containing different user limits,
              corresponds to "u limit" command.
```

1.150 $bu_access

```
                  Variable   $bu_access

       Type   number

Description   Returns a number containing differnt user acces bits.

              The bits are:
                0 DOS
                1 Conferences
                2 Files
                3 Private messages
                4 Passwords
               30 Download
               31 Upload
              255 Sysop

   See also
                Bitwise Operation
```

1.151 $bu_utoggles

```
                  Variable   $bu_utoggles

       Type   number

Description   Returns a number containing several different user toggles.

              The bits are:
               0 Insert mode
               1 Indent mode
               2 XY Display in editor
               3 Don't flash your name
               4 Conference status at login
               5 Expert mode
               6 G&R commands
               7 Colors
               8 Review own messages
               9 Real VT100 keyboard
              10 Quote messages
              11 Silent mode

   See also
                Bitwise Operation
```

1.152 $bu_lasttime

```
                  Variable   $bu_lasttime

       Type   number

Description   Returns a datecode for the last time the current user
```

```
              logged on to the board.

              Bits  Meaning
              ====  ====================
              0-4   Seconds/2
              5-10  Minutes
              11-15 Hours
              16-20 Day
              21-26 Month
              25-31 Year-1980

   See also
                Bitwise Operation
```

1.153 $bu_msgread

```
   Variable   $bu_msgread

       Type   number

Description   Returns the number of total messages read for the
              current user.
```

1.154 $bu_msgdumped

```
   Variable   $bu_msgdumped

       Type   number

Description   Returns the number of total messages dumped for the
              current user.
```

1.155 $bu_bytesup

```
   Variable   $bu_bytesup

       Type   number

Description   Returns the total number of bytes uploaded for the
              current user.
```

1.156 $bu_bytesdown

```
   Variable   $bu_bytesdown

       Type   number
```

```
Description   Returns the total number of bytes downloaded for the
              current user.
```

1.157 $bu_userbits

```
                   Variable   $bu_userbits

       Type   number

Description   Returns a number contianing userbits. This is not used
              by BBBS so feel free to use this variable as you want, but
              remember that others might use it as well.

   See also
                Bitwise Operation
```

1.158 Internal variables

```
                         SUBTOPICS:

                  $bv_filna

                  $bv_local_buffe

                  $bv_comhandle

                  $bv_comdevice

                  $bv_comstring

                  $bv_interface

                  $bv_crrdir

                  $bv_holdreal

                  $bv_hddesc

                  $bv_serna

                  $bv_tempsys

                  $bv_unum

                  $bv_confnro

                  $bv_baud

                  $bv_realbaud

                  $bv_loginhour
```

```
$bv_loginmin

$bv_lastmsg

$bv_curmsg

$bv_firstmsg

$bv_lastreaded

$bv_newavail

$bv_serno

$bv_nodenumber

$bv_reduced

$bv_nextevent

$bv_temptime

$bv_com

$bv_realmnp

$bv_carrier

$bv_outputstopp

$bv_quicklogin

$bv_paged

$bv_nobreaknow

$bv_groups

$bv_txt
```

1.159 $bv_filna

```
   Variable   $bv_filna

       Type   string

Description   Contains the name of the last processed file.

Example code  function main() {
                printf("Last processed file: %s\n",$bv_filna);
              }
```

1.160 $bv_local_buffe

```
   Variable   $bv_local_buffe

       Type   string

Description   This is the keyboard buffer used by BBBS in all input
              routines.

Example code  function main() {
                $bv_local_buffe = "G;Y;";
              }
```

1.161 $bv_comhandle

```
   Variable   $bv_comhandle

       Type   number (constant)

Description   Has communication device handle (OS/2).

Example code function main() {
               system(sprintf("foo.exe %u %s",$bv_comhandle,$bv_comdevice),0);
              }
```

1.162 $bv_comdevice

```
   Variable   $bv_comdevice

       Type   string (constant)

Description   Has communication device name (OS/2).

Example code function main() {
               system(sprintf("foo.exe %u %s",$bv_comhandle,$bv_comdevice),0);
              }
```

1.163 $bv_comstring

```
   Variable   $bv_comstring

       Type   string

Description   Contains the command queue used by BBBS.

Example code  function main() {
                if (pos("TEST",$bv_comstring)) // if queue contains TEST
                  $bv_comstring = "";          // then delete queue
              }
```

1.164 $bv_interface

```
   Variable   $bv_interface

       Type   number

Description   Is used as a indicator which identifies b-mode or not.

Example code  function main() {
                if (!$bv_interface)
                  printf("Standard mode is active\n");
                else
                  printf("You are currently in b-mode\n");
              }
```

1.165 $bv_crrdir

```
   Variable   $bv_crrdir

       Type   string

Description   Returns the current dir in BBBS's internal File/4 system.

Example code  function main() {
                printf("Current dir: %s\n",$bv_crrdir);
              }
```

1.166 $bv_holdreal

```
   Variable   $bv_holdreal

       Type   string

Description   Returns the path to current node's hold directory.

Example code  function main() {
                if (fopen(0,strcat($bv_holdreal,"windows.bin"),"rb")) {
                  fclose(0);
                  printf("DANGER! Will Robinson DANGER! Alien lifeform!\n");
                }
              }
```

1.167 $bv_hddesc

```
   Variable   $bv_hddesc

       Type   string

Description   Returns the path and filename of the current node's
```

```
              descript.ion file containing the descriptions for files in
              hold.

Example code  function main() {
                if (fopen(0,$bv_hddesc,"rt")) {
                  fclose(0);
                  printf("You have files in your hold-area!\n");
                }
              }
```

1.168 $bv_serna

```
   Variable   $bv_serna

       Type   string

Description   Returns the registered BBBS Name from BBBS.KEY

Example code  function main() {
                printf("You have reached %s\n",$bv_serna)
              }
```

1.169 $bv_tempsys

```
   Variable   $bv_tempsys

       Type   number

Description   Change the temporary SysOp level for the current user.
              SysOp's access is a bitfield integer. You can use values from
              0 to 255, as following:

               1 Can shell to OS and execute OS commands
               2 Full access to all conferences
               4 Full access to all files
               8 May read private messages from all conferences
              16 May change passwords
              32 May edit user's status (kill, status)

              To give a certain access just add the numbers. For example, if
              you want a user to have access to all conferences and  private
              messages, the value is 10 (2+8).

Example code  function main() {
                var $name, $passw, $old_sys;

                if ($bu_name == "FOO USER") {
                  printf("You may change the password on a user.\n");
                  if (($name = input("User        : ",80,0))!="") {
                    $passw = input("New password: ",10,1);
                    $old_sys = $bv_tempsys;
                    $bv_tempsys = 16;
```

```
                bbbs(sprintf("q;u;find;\"%s\";p;%s;%s;;",$name,
                                                        $passw,
                                                        $name);
                $bv_tempsys = $old_sys;
              }
            }
          }
```

1.170 $bv_unum

```
   Variable   $bv_unum

       Type   number

Description   Returns the current users user number from the user
              database.

Example code  function main() {
                printf("You are user number %u\n",$bv_unum);
              }
```

1.171 $bv_confnro

```
   Variable   $bv_confnro

       Type   number

Description   Returns the number of the current conference.

Example code  function main() {
                printf("You are in conference number %u\n",$bv_confnro);
              }
```

1.172 $bv_baud

```
   Variable   $bv_baud

       Type   number

Description   Returns the baudrate for the current call. If it is a
              local login, this number will be 9600.
```

1.173 $bv_realbaud

```
   Variable   $bv_realbaud

       Type   number
```

```
Description   Returns the real baudrate for the current call. If it
              is a local login, this number will be 0.
```

1.174 $bv_loginhour

```
                   Variable   $bv_loginhour

       Type   number

Description   Returns the hour part of the time when the user
              logged in.

Example code  function main() {
                printf("You logged in at %02u:%02u\n",$bv_loginhour,

                $bv_loginmin
                );
              }
```

1.175 $bv_loginmin

```
                   Variable   $bv_loginmin

       Type   number

Description   Returns the minute part of the time when the user
              logged in.

Example code  function main() {
                printf("You logged in at %02u:%02u\n",
                $bv_loginhour
                ,
                                                    $bv_loginmin);
              }
```

1.176 $bv_lastmsg

```
   Variable   $bv_lastmsg

       Type   number

Description   Returns the number of last message read in current
              conference.

Example code  function main() {
                printf("Last message read: %u\n",$bv_lastmsg);
              }
```

1.177 $bv_curmsg

```
   Variable   $bv_curmsg

       Type   number

Description   Returns current message number.

Example code  function main() {
                printf("Current message number is: %u\n",$bv_curmsg);
              }
```

1.178 $bv_firstmsg

```
   Variable   $bv_firstmsg

       Type   number

Description   Returns number of first available message in current
              conference.

Example code  function main() {
                printf("First message number is: %u\n",$bv_firstmsg);
              }
```

1.179 $bv_lastreaded

```
   Variable   $bv_lastreaded

       Type   number

Description   Returns number of last read message in current
              conference.

Example code  function main() {
                printf("Last message read is: %u\n",$bv_lastreaded);
              }
```

1.180 $bv_newavail

```
   Variable   $bv_newavail

       Type   number

Description   Returns a number indicating how many new messages the
              current conference contains.

Example code  function main() {
                printf("You have %u new messages.\n",$bv_newavail);
              }
```

1.181 $bv_serno

```
   Variable   $bv_serno

       Type   number

Description   Returns the BBBS registration number from BBBS.KEY.

Example code  function main() {
                printf("Serial number: %u\n",$bv_serno);
              }
```

1.182 $bv_nodenumber

```
   Variable   $bv_nodenumber

       Type   number

Description   Returns the nodenumber for the current node.

Example code  function main() {
                printf("You are connected to node %u\n",$bv_nodenumber);
              }
```

1.183 $bv_reduced

```
   Variable   $bv_reduced

       Type   number

Description   Returns the number of minutes the current users
              timelimit has been reduced by due to an event.

Example code  function main() {
                printf("Timelimit is reduced by %u minutes\n",$bv_reduced);
              }
```

1.184 $bv_nextevent

```
   Variable   $bv_nextevent

       Type   number

Description   Returns the number of minutes left to the next event.

Example code  function main() {
                printf("Next event is due in %u minutes\n",$bv_nextevent);
              }
```

1.185 $bv_temptime

```
   Variable   $bv_temptime

       Type   number

Description   Returns the timelimit for the current call.

Example code  function main() {
                printf("You have %u min. left for this call.\n",$bv_temptime);
              }
```

1.186 $bv_com

```
   Variable   $bv_com

       Type   number

Description   Returns the COM port number of current node.

Example code  function main() {
                printf("You are running on COM port %u\n",$bv_com);
              }
```

1.187 $bv_realmnp

```
   Variable   $bv_realmnp

       Type   number

Description   Toggles whether real MNP is active or not.

              0 MNP not active
              1 MNP active
```

1.188 $bv_carrier

```
   Variable   $bv_carrier

       Type   number

Description   Inidicates the state of modem carrier

              0 inactive
              1 active

              Can be used to decide if the user hangs up or not.
              SHOULD be used when running loops in scripts,
              BBBS will not abort the script even if user hangs up.
```

```
Example code  function main() {
                while ((!kbhit())&&($bv_carrier))
                printf("Hit any key (or hangup)!!!\n");
              }
```

1.189 $bv_outputstopp

```
   Variable   $bv_outputstopp

       Type   number

Description   (Dis)enables output to the user. Can be used when you
              want to do something which the user shouldn't see.

              ! BBBS will set this to 1 after a --more-- listing, but reset
              it when it comes to the ordinary prompt. Should be reset in
              scripts after listings.

Example code  function main() {
                printf("Please wait while joining some conferences..\n");
                $bv_outputstopp = 1;
                bbbs("j;bbbs.english;");
                bbbs("j;bbbs.util;");
                $bv_outputstopp = 0;
              }
```

1.190 $bv_quicklogin

```
   Variable   $bv_quicklogin

       Type   number

Description   Toggles whether quicklogin was used or not.

              0 not used
              1 used
```

1.191 $bv_paged

```
   Variable   $bv_paged

       Type   number

Description   Number of times user has requested chat with sysop.
```

1.192 $bv_nobreaknow

```
   Variable   $bv_nobreaknow

       Type   number

Description   Allow user to break screen output with C-c.
```

1.193 $bv_groups

```
   Variable   $bv_groups

       Type   string (constant)

Description   Returns a string with all the groups the current user
              are a member of. All groups are seperated by a comma (,).
```

1.194 $bv_txt

```
   Variable   $bv_txt[linenumber]

       Type   string (constant) [array]

Description   Returns the text at the linenumber you supply from
              BBBSTXTx where x is nothing for english or the language code
              the current user is using.

Example code  Using english language the following command:
              printf("%s\n",$bv_txt[13]);
              will output:
              Sorry, no resume info in this system.
```

1.195 Variable list

Variable name	Type	Constant
$bu_name	string	*
$bu_address	string	
$bu_city	string	
$bu_phone	string	
$bu_birth	string	
$bu_ok2login	number	
$bu_termcap	number	
$bu_pagelength	number	
$bu_charset	number	
$bu_language	number	
$bu_readmode	number	
$bu_packtype	number	
$bu_protocol	number	

```
$bu_nodemsgfilt         number
$bu_timelimit           number
$bu_timeleft            number
$bu_timeson             number
$bu_msgleft             number
$bu_uploaded            number
$bu_downloaded          number
$bu_pmsgleft            number
$bu_pfileup             number
$bu_pfiledown           number
$bu_pcalled             number
$bu_fchecked            number
$bu_timebank            number
$bu_resume              number
$bu_limits              number
$bu_access              number
$bu_utoggles            number
$bu_lasttime            number
$bu_msgread             number
$bu_msgdumped           number
$bu_bytesup             number
$bu_bytesdown           number
$bu_userbits            number
$bc_count_of_co         number          *
$bc_post_conf           number          *
$bc_resume_conf         number          *
$bc_fileinfo_co         number          *
$bc_lastread            number          *
$bc_status              number          *
$bc_confname            string          *
$bc_description         string          *
$bg_bbbs_name           string          *
$bg_sysop_name          string          *
$bg_closed_pass         string          *
$bg_grabfile            string          *
$bg_maindir             string          *
$bg_updir               string          *
$bg_tempdir             string          *
$bg_menudir             string          *
$bg_tickdir             string          *
$bg_netmail             string          *
$bg_inbound             string          *
$bg_max_nodes           number          *
$bg_bankmax             number          *
$bg_newu_time           number          *
$bg_bankrate            number          *
$bg_msgrate             number          *
$bg_newu_access         number          *
$bg_gtoggles            number          *
$bl_modem_init1         string          *
$bl_modem_init2         string          *
$bl_modem_init3         string          *
$bl_modem_hangu         string          *
$bl_modem_busy_         string          *
$bl_modem_answe         string          *
$bl_fd_dobbs            string          *
$bl_logfile             string          *
```

```
$bl_loginlog            string          *
$bl_grabdir             string          *
$bl_menudir             string          *
$bl_newdir              string          *
$bl_start_speed         number          *
$bl_min_speed           number          *
$bl_base_addres         number          *
$bl_irq                 number          *
$bl_pollrate            number          *
$bl_answer_ring         number          *
$bl_ltoggles            number          *
$bv_filna               string
$bv_local_buffe         string
$bv_comhandle           number
$bv_comdevice           string          *
$bv_comstring           string          *
$bv_interface           number
$bv_crrdir              string
$bv_holdreal            string
$bv_hddesc              string
$bv_serna               string          *
$bv_tempsys             number
$bv_unum                number          *
$bv_confnro             number          *
$bv_baud                number
$bv_realbaud            number
$bv_loginhour           number
$bv_loginmin            number
$bv_lastmsg             number
$bv_curmsg              number
$bv_firstmsg            number
$bv_lastreaded          number
$bv_newavail            number
$bv_serno               number          *
$bv_nodenumber          number          *
$bv_reduced             number
$bv_nextevent           number
$bv_temptime            number
$bv_com                 number          *
$bv_realmnp             number
$bv_carrier             number
$bv_outputstopp         number
$bv_quicklogin          number
$bv_paged               number
$bv_nobreaknow          number
$bv_groups              string          *
$bv_txt                 string          *
$bn_split               number
$bn_bstatus             number
$bn_zstatus             number
$bn_speed               number
$bn_time                number
$bn_name                string
$bn_realname            string
$bf_timleft             number          *
$bf_hour                number          *
$bf_min                 number          *
```

```
$bf_sec                 number          *
$bf_year                number          *
$bf_month               number          *
$bf_day                 number          *
```

1.196 Bitwise operation

```
Logic table for all three bitwise operations:

      |  and  |   or  |  xor
x  y  | x & y | x | y | x ^ y
------+-------+-------+------
0  0  |   0   |   0   |   0
0  1  |   0   |   1   |   1
1  0  |   0   |   1   |   1
1  1  |   1   |   1   |   0


Other important commands are >> and <<, the (bitwise) shift
operations. << shifts bits left and >> shifts them right.

For example:

2 << 1 == 4 (binary 0010 shifted one bit left is 0100)
4 >> 2 == 1 (binary 0100 shifted two bits right is 0001)


So, when you want to check bit 8 from variable, you can do it
like this:  (variable >> 8) & 1

To set bit 8 of variable:  variable | (1 << 8)

To clear bit 8 of variable:  variable & (-1 ^ (1 << 8))

To get enum of bits 8-10 from variable:  (variable >> 8) & 7
```

1.197 BBBS Scripts

```
BBBS runs following programs:

        ASCRIPT         always when user logs in
        BROBO           when user sends an unknown command to BRoboCop
        DSCRIPT         after a user has downloaded a file
        ERROR           after invalid command, all menus
        GSCRIPT         when user logges out normally
        LSCRIPT         when user logges out, after GSCRIPT
        OPENDOOR        by main command `OPen`
        RSCRIPT         after a new user has registered
        SCRIPT          by main command `Answer`
        USCRIPT         after a user has uploaded a file
        HOTLOG          after a special hotlogin string from user
        GOTFAX          after recieving a fax
```

```
GOTVOICE        after recieving a voice call
```

1.198 How to use the help system

```
The help system commands:

Contents  - Shows the contents (the main help menu)
Index     - Shows an alpateical index
Help      - Shows you this help
Retrace   - Shows the last helpscreen you looked at
Browse <  - Shows the previous helpscreen in the helpfile
Browse >  - Shows the next helpscreen in the helpfile
Quit      - Quits back to BBBS

When you are in the helpsystem you can move up and down with
arrow keys. The keys Ctrl-R and Ctrl-C will scroll one page Up
or Down.

Often you will see words or sentences that are marked with a
different background color. These are links to other related
information. Just press enter when you have selected a link to
get more information about that subject.
```