

AML.HYPER

Andy Grifo

COLLABORATORS

	TITLE : AML.HYPER		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Andy Grifo	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AML.HYPER	1
1.1	MAIN	1
1.2	AML_FORMAT	1
1.3	WHATIS_AML	1
1.4	AML_LIST	2
1.5	FAST_AML	2
1.6	BOOL	3
1.7	TC	3
1.8	BC	3
1.9	CLS	4
1.10	BOLD	4
1.11	ANSIRESET	5
1.12	CURSOR	5
1.13	WAITKEY	5
1.14	CURSOR_DIRECTION	6
1.15	UNDERLINE	6
1.16	RVIDEO	7
1.17	ITALICS	7
1.18	CPOS	7
1.19	NL	8
1.20	REVIEW	8
1.21	YPOS	9
1.22	FNAME	9
1.23	LNAME	10
1.24	ANSI	10
1.25	STRWID	11
1.26	NAME	12
1.27	UNAME	12
1.28	VOICENO	13
1.29	DATANO	13

1.30	PASSWORD	14
1.31	HANDLE	14
1.32	ADDRESS	14
1.33	CITY	14
1.34	PROVINCE	15
1.35	POSTCODE	15
1.36	COUNTRY	15
1.37	LEVEL	16
1.38	VALID	16
1.39	CALLS	16
1.40	MSGLEFT	16
1.41	MSGREC	17
1.42	FILESUP	17
1.43	FILESDN	17
1.44	BYTESUP	18
1.45	BYTESDN	18
1.46	CREDITS	18
1.47	FRATIO	18
1.48	BRATIO	19
1.49	SCRLENGTH	19
1.50	SCRWIDTH	19
1.51	MORE	20
1.52	TOTALFILES	20
1.53	AREANUMBER	20
1.54	FILEPOS	20
1.55	AREANAME	21
1.56	INDEXSORT	21
1.57	DIRECTION	21
1.58	FILTER	22
1.59	FILESMARKED	22
1.60	RAW	22
1.61	ECHO	23
1.62	CRLF	23
1.63	CTRL_C	23
1.64	CTRL_D	24
1.65	HANDLER	24
1.66	COL	25

Chapter 1

AML.HYPER

1.1 MAIN

ApT-BBS AML Guide

Copyright (c) 1995 ApT-Design

```
~AML~List~~~~~  
~What~is~AML~exactly?~~~~~  
~Formatting~AML~Codes,~sprintf~style~~~~~  
~Faaaassst~AML~~~~~
```

1.2 AML_FORMAT

AML Codes that return strings can all be formatted into a FORCED width by using the ~STRWID~ command. This enables text to be set to a forced width, resulting in all subsequent text strings returned being the same width.

This is a very handy feature and will be of much use once it's power is fully understood.

1.3 WHATIS_AML

AML Is a sequence of SWITCH codes, normally denoted by the percentage sign % which tells ApT-BBS that it is to do something depending on the command tjay was embedded within the % , leading upto the ending/closing dot . sign.

% denotes the START of a AML specific command
. denotes the ENDING of a AML specific command

As a small example - if you have a file of some form that contains text that is being displayed to the screen, a menu of sorts. Then you may wish to have the users last name displayed prior to asking

the user for a keypress. Embedding the following within the menu,

```
"Well then %FNAME., time to press a key don't you think =>"
```

Would have the effect of being converted into the following line

```
Well then Andy, time to press a key don't you think =>
```

AML is built within most of the ApT-BBS display system, ranging from commands such as the AREXX 'MSG' command to the programmers library ShowFile("filename"); command, even menus that are displayed. Just about everything that is displayed passes through the AML system at some point.

AML is so powerful and fast that it even checks to see if users have the ability to take various commands. Items that change the colour are of no use to users who have no ANSI abilities.

AML also is so powerful in that it monitors the actual RAW escape codes, the type of which are located within ANSI editors that save out screen designs. Some users may have preferenced NO ANSI and AML removes items of a unwanted nature.

1.4 AML_LIST

ApT-BBS AML List

```
~TC~~~~~ ~BC~~~~~ ~CLS~~~~~ ~BOLD~~~~~ ~ANSIRESET~~~~~
~CURSOR~~~~~ ~WAITKEY~~~~~ ~CURSOR_LEFT~~ ~CURSOR_RIGHT~ ~CURSOR_UP~~~~~
~CURSOR_DOWN~~ ~UNDERLINE~~~~~ ~RVIDEO~~~~~ ~ITALICS~~~~~ ~CPOS~~~~~
~NL~~~~~ ~REVIEW~~~~~ ~YPOS~~~~~ ~FNAME~~~~~ ~LNAME~~~~~
~ANSI~~~~~ ~LOCKSERIAL~~~ ~STRWID~~~~~ ~NAME~~~~~ ~UNAME~~~~~
~VOICENO~~~~~ ~DATANO~~~~~ ~PASSWORD~~~~~ ~HANDLE~~~~~ ~ADDRESS~~~~~
~CITY~~~~~ ~PROVINCE~~~~~ ~POSTCODE~~~~~ ~COUNTRY~~~~~ ~LEVEL~~~~~
~VALID~~~~~ ~CALLS~~~~~ ~MSGLEFT~~~~~ ~MSGREC~~~~~ ~FILESUP~~~~~
~FILES DN~~~~~ ~BYTESUP~~~~~ ~BYTESDN~~~~~ ~CREDITS~~~~~ ~FRATIO~~~~~
~BRATIO~~~~~ ~SCRLLENGTH~~~~~ ~SCRWIDTH~~~~~ ~MORE~~~~~
```

The following are only valid for use within a initiated file-area, and loaded file-index area.

```
~TOTALFILES~~~ ~AREANUMBER~~~ ~FILEPOS~~~~~ ~AREANAME~~~~~ ~INDEXSORT~~~~~
~DIRECTION~~~~~ ~FILTER~~~~~ ~FILESMARKED~~
```

The following are for use as additional arguments, for such things as calling DOS Doors.

```
~RAW~~~~~ ~ECHO~~~~~ ~CRLF~~~~~ ~CTRL_C~~~~~ ~CTRL_D~~~~~
~HANDLER~~~~~ ~COL~~~~~
```

1.5 FAST_AML

The following commands are the fastest draw on the west when it comes to changing a style. When speed is of the essence these commands come into their own.

All of the commands below start with a different switch character than the standard set of AML Codes, this is the \ key.

```
\c<number>  Changes the Text colour 0-9          \c7 WHITE TEXT
\z<number>  Changes the Background colour 0-9      \z4 BLUE BACKGROUND

\oBOOL      Changes the Cursor, ON/OFF             \o0 Cursor OFF \uBOOL
Changes Underline mode ON/OFF                      \u1 Underline ON \bBOOL      Changes
Bold mode ON/OFF                                   \b1 BOLD ON \iBOOL      Changes Italic mode
ON/OFF                                              \i0 ITALICS OFF

\ @          Clears the screen, if preferenced to do so. \n      Newline
character.
```

1.6 BOOL

BOOLEAN values within the AML instances should always be specified as either 0 or 1, 0 denotes FALSE and 1 denotes TRUE (OFF/ON)

1.7 TC

TITLE:

Text Colour

TEMPLATE:

```
%TC=<colour>.
```

FUNCTION:

Changes the Text colour to one of the selected colour sequences.

EXAMPLE:

```
%TC=1.
```

Changes the Text colour to Red.

1.8 BC

TITLE:

Background Colour

TEMPLATE:

%BC=<colour>.

FUNCTION:

Changes the Background colour to one of the selected colour sequences.

EXAMPLE:

%BC=7.

Changes the Background colour to White.

1.9 CLS

TITLE:

Clear the screen.

TEMPLATE:

%CLS.

FUNCTION:

Clears the screen using two forms of Screen clear code, thus making quite sure the screen is refreshed on all terminal software.

The screen will only be cleared if the user has screen clear codes selected within his preferences. If you are required to make sure the user actually has his screen cleared, then a hard-coded clear code may be required.

1.10 BOLD

TITLE:

Bold - Toggle.

TEMPLATE:

%BOLD=BOOL.

FUNCTION:

Turns BOLD lettering ON or OFF depending on which of the boolean switches is passed.

EXAMPLE:

%BOLD=1. Turns BOLD lettering ON

`%BOLD=0.` Turns BOLD lettering OFF

1.11 ANSIRESET

TITLE:

Ansi Reset

TEMPLATE:

`%ANSIRESET.`

FUNCTION:

Quite often it is required for all of the Styles to be turned off quickly and easily. This command allows just that. It is generally recommended to turn on a default colour after use of this call, as on some communication packages the default text (after the reset) will be RED.

1.12 CURSOR

TITLE:

Toggle Cursor On/Off

TEMPLATE:

`%CURSOR=BOOL.`

FUNCTION:

Toggles the Cursor On/Off - This speeds up the displaying of text quite considerably. It should be noted that a few terminal packages can not handle the turning on and off of the cursor, for some odd reason, and thus this option is "preferenced" and stored.

EXAMPLE:

`%CURSOR=0.`

1.13 WAITKEY

TITLE:

Waitkey

TEMPLATE:

%WAITKEY.

FUNCTION:

Waits for a key to be pressed by the user - Can be embedded within a string so that the remaining text, if placed within the middle, is displayed after the key-press.

EXAMPLE:

"Gosh your slow, .. how about pressing a key [Waiting]%WAITKEY.Thanks!!"

1.14 CURSOR_DIRECTION

TITLE:

Cursor Direction

TEMPLATE:

%CURSOR_LEFT=<positions>.
%CURSOR_RIGHT=<positions>.
%CURSOR_UP=<positions>.
%CURSOR_DOWN=<positions>.

FUNCTION:

Moves the cursor within the direction pertaining to the command, so many spaces depending on the positions supplied.

EXAMPLE:

"I would like to teach the world sing..%CURSOR_LEFT=9.to"

"I would like to teach the world to sing"

1.15 UNDERLINE

TITLE:

Underline

TEMPLATE:

%UNDERLINE=BOOL.

FUNCTION:

Turns underline mode on/off

EXAMPLE:

```
%UNDERLINE=1. Turns underline mode on.
```

1.16 RVIDEO

TITLE:

```
Reverse Video
```

TEMPLATE:

```
%RVIDEO=BOOL.
```

FUNCTION:

```
Turns Reverse-Video mode on/off
```

EXAMPLE:

```
%RVIDEO=0. Turns Reverse Video mode off.
```

1.17 ITALICS

TITLE:

```
Italics Style
```

TEMPLATE:

```
%ITALICS=BOOL.
```

FUNCTION:

```
Turns the Italics Style on/off.
```

EXAMPLE:

```
%ITALICS=1. Turns Italics style on.
```

1.18 CPOS

TITLE:

```
Cursor Position
```

TEMPLATE:

```
%CPOS=<ypos,xpos>.
```

FUNCTION:

Positions the cursor at the required position pertaining to the Y and X co-ordinates given.

EXAMPLE:

"%CPOS=12,13. Hello World, I am at Y Position 12, and X Position 13"

1.19 NL

TITLE:

Newline

TEMPLATE:

%NL=<number of new lines>

%NL.

FUNCTION:

Displays either one, or a number of newlines.

EXAMPLE:

"pretend-start%NL=3.pretend-end."

pretend-start

pretend-end

"This is line 1 %NL. This is Line 2 %NL. This is Line 3 %NL."

This is line 1
This is line 2
This is line 3

1.20 REVIEW

TITLE:

Review Date

TEMPLATE:

%REVIEW.

FUNCTION:

Displays the date when the user comes up, next, for review.

EXAMPLE:

".. and your account comes up for review on %REVIEW."

.. and your account comes up for review on 12-Aug-94

1.21 YPOS

TITLE:

Y-Position More counter.

TEMPLATE:

%YPOS=<yposition>

FUNCTION:

The users more counter (termed the Y Counter) reaches a stage, the users maximum length of screen, whereby the user is asked if they would like the remainder of the text displayed. This function allows you to set the counter at a position of your choice, which can be handy in certain instances.

1.22 FNAME

TITLE:

First Name.

TEMPLATE:

%FNAME.

FUNCTION:

Display the users FIRST name.

EXAMPLE:

If the users name was "Joe Bloggs" then having,

"Your first name is %FNAME., Isn't it??"

would result in,

Your first name is Joe, isn't it??"

1.23 LNAME

TITLE:

Last Name

TEMPLATE:

Displays the users LAST name.

FUNCTION:

If the users name was "Joe Bloggs" then having,

"Your last name is %LNAME., Isn't it??"

would result in,

Your first name is Bloggs, isn't it??"

EXAMPLE:

1.24 ANSI

TITLE:

Temporary Ansi Setting.

TEMPLATE:

%ANSI.

%ANSI=BOOL.

FUNCTION:

This command turns the ANSI flags on/off depending on the command used.

EXAMPLE:

%ANSI. Results within the Ansi flags being turned ON.

%ANSI=0. Turns the users ansi flags off..

%ANSI=1. Turns the users ansi flags on..

NOTE:

This will only be a temporary setting of the users ansi flags, for the duration of the display in hand, be it text-file or otherwise. The users actual "preferenced" settings will be re-invoked after the actions have been carried out (after the file is displayed, or the text displayed..)

1.25 STRWID

TITLE:

String Width

TEMPLATE:

%STRWID=<Width>.

FUNCTION:

This is a VERY handy function when using any form of AML code that returns a STRING, such as the users name or telephone number, file area information (area name, credits, etc.. etc..) The reason it is handy is because normally if you did the following (within a menu template file, and the speech-marks are needed for this one example)

"%FNAME."

would display

"Sir_Andrew" ('Sir' to make the name longer of-course :)

If indeed you had another user, one might get:

"Dave" - 4 Characters wide
 "Steve" - 5 Characters wide
 "Tom" - 3 Characters wide

The thing to note here is that all the returned strings have different widths. This can be rather worrying if one wishes to format such strings within a IMB border frame, of sorts. So, in order to force the name to be a set WIDTH (no matter what length the name is..) one would use, prior to the accessing of the string, the command

%STRWID=10.

Which would result in all the above being displayed as:

"Sir_Andrew" - 10 Characters wide
 "Dave " - 10 Characters wide
 "Steve " - ... and so on,
 "Tom " - ... and so forth

All 10 characters in width. One could also use it in the other form,

%STRWID=5."%FNAME."

"Sir_A" - 5 Characters wide
 "Dave " - etc..
 "Steve"
 "Tom "

This allows for you to correctly format your menus, but remember, you need to place the %STRWID=<width>. BEFORE the accessing of the required variable, just to initiate the value (if not already done

so)

You only need to specify the %STRWID=<width>. at least ONCE and all returned strings will be THAT width. You should also turn the function off when not required by giving a value of 0, as in %STRWID=0. (Turns the function off, and all strings are the normal width)

As a short example, the file/fmenu.tpl file normally displays information on the file-area statistics for the area the user is within. In order to do this we require the use of the new AML codes, and to also correctly format the menu within the IBM border we would use the following, within the required portion of the file.

```
%STRWID=14.
.------.
| Area Number: %AREANUMBER.|
| Area Name   : %AREANAME.|
| Total Files: %TOTALFILES.|
| Index Sort  : %INDEXSORT.|
| Filter      : %FILTER.|
| Direction   : %DIRECTION.|
| Position    : %FILEPOS.|
| Credits     : %CREDITS.|
\-----/
%STRWID=0.
```

This would display the above box correctly formatted.. Most powerful, infact, most handy.

1.26 NAME

TITLE:

Name Display

TEMPLATE:

%NAME.

FUNCTION:

Displays the full name of the user.

EXAMPLE:

If the name of the user online is "Joe Bloggs" then the command would return that as the name.

1.27 UNAME

TITLE:

Underline Name

TEMPLATE:

%UNAME.

FUNCTION:

This has the effect of returning the users full name, but places underscores '_' inplace of all spaces that would otherwise be returned.

EXAMPLE:

"I bet you didn't know that your name, underscored, is %UNAME."

I bet you didn't know that your name, underscored, is Joe_Bloggs.

1.28 VOICENO

TITLE:

Voice number

TEMPLATE:

%VOICENO.

FUNCTION:

Returns the entry within the users stats that relate to the voice number entrance.

1.29 DATANO

TITLE:

Data number

TEMPLATE:

%DATANO.

FUNCTION:

Returns the users Data telephone number.

1.30 PASSWORD

TITLE:

Password

TEMPLATE:

%PASSWORD.

FUNCTION:

Returns the users password.

1.31 HANDLE

TITLE:

Handle/Alias

TEMPLATE:

%HANDLE.

FUNCTION:

Returns the users Handle/Alias.

1.32 ADDRESS

TITLE:

Address

TEMPLATE:

%ADDRESS.

FUNCTION:

Returns the first portion of the users address.

1.33 CITY

TITLE:

City

TEMPLATE:

%CITY.

FUNCTION:

Returns the portion of the users address relating to the City.

1.34 PROVINCE

TITLE:

Province

TEMPLATE:

%PROVINCE.

FUNCTION:

Returns the portion of the users address relating to the province.

1.35 POSTCODE

TITLE:

Postcode

TEMPLATE:

%POSTCODE.

FUNCTION:

Returns the portion of the users address relating to the Postal/Zip code.

1.36 COUNTRY

TITLE:

Country

TEMPLATE:

%COUNTRY.

FUNCTION:

Returns the Country, 3 characters normally.

1.37 LEVEL

TITLE:

Access Level

TEMPLATE:

%LEVEL.

FUNCTION:

Returns the access level of the user.

1.38 VALID

TITLE:

Valid

TEMPLATE:

%VALID.

FUNCTION:

Returns BOOL state of the users validity.

1.39 CALLS

TITLE:

Calls

TEMPLATE:

%CALLS.

FUNCTION:

Returns the number of calls the user has made to the system.

1.40 MSGLEFT

TITLE:

Messages Left

TEMPLATE:

%MSGLEFT.

FUNCTION:

Returns the number of messages the user has left.

1.41 MSGREC

TITLE:

Messages received

TEMPLATE:

%MSGREC.

FUNCTION:

Returns the number of messages the user has received.

1.42 FILESUP

TITLE:

Files uploaded

TEMPLATE:

%FILESUP.

FUNCTION:

Returns the number of files the user has uploaded.

1.43 FILESDN

TITLE:

Files Downloaded

TEMPLATE:

%FILESDN.

FUNCTION:

Returns the number of files the user has Downloaded.

1.44 BYTESUP

TITLE:

Bytes Uploaded

TEMPLATE:

%BYTESUP.

FUNCTION:

Returns the number of Bytes files the user has Uploaded.

1.45 BYTESDN

TITLE:

Bytes Downloaded

TEMPLATE:

%BYTESDN.

FUNCTION:

Returns the number of Bytes the user has Downloaded.

1.46 CREDITS

TITLE:

Credits

TEMPLATE:

%CREDITS.

FUNCTION:

Returns the number of Credits the user has left before they are required to upload.

1.47 FRATIO

TITLE:

File Ratio

TEMPLATE:

%FRATIO.

FUNCTION:

Returns the File-Ratio that the user has.

1.48 BRATIO

TITLE:

Bytes Ratio

TEMPLATE:

%BRATIO.

FUNCTION:

Returns the Bytes-Ratio the user has.

1.49 SCRLENGTH

TITLE:

Screen Length

TEMPLATE:

%SCRLENGTH.

FUNCTION:

Returns the Length of the screen the user has preferenced.

1.50 SCRWIDTH

TITLE:

Screen Width

TEMPLATE:

%SCRWIDTH.

FUNCTION:

Returns width of the screen that the user has preferenced.

1.51 MORE

TITLE:

More Prompt

TEMPLATE:

%MORE.

FUNCTION:

Returns state of the users More-Prompt variable.

1.52 TOTALFILES

TITLE:

Total Files

TEMPLATE:

%TOTALFILES.

FUNCTION:

Returns the total amount of files that are within the current file-area.

1.53 AREANUMBER

TITLE:

File Area Number

TEMPLATE:

%AREANUMBER.

FUNCTION:

Returns the number of the file-area that the user is currently within.

1.54 FILEPOS

TITLE:

File Position

TEMPLATE:

%FILEPOS.

FUNCTION:

Returns the position within the current file-index that the user is currently positioned on.

1.55 AREANAME

TITLE:

File Area Name

TEMPLATE:

%AREANAME.

FUNCTION:

Returns the name of the file-area that the user is currently within.

1.56 INDEXSORT

TITLE:

Index Sort

TEMPLATE:

%INDEXSORT.

FUNCTION:

Returns the form of Index sorting that has been performed within the current file-index.

1.57 DIRECTION

TITLE:

File Scanning Direction

TEMPLATE:

%DIRECTION.

FUNCTION:

Returns the direction that the file-scanning is being done within.

1.58 FILTER

TITLE:

File Filter

TEMPLATE:

%FILTER.

FUNCTION:

Returns the state of the File-Filter setting.

1.59 FILESMARKED

TITLE:

Files Marked

TEMPLATE:

%FILESMARKED.

FUNCTION:

Returns the number of files that the user has marked for download.

1.60 RAW

TITLE:

Raw Mode

TEMPLATE:

%RAW.

FUNCTION:

When requesting for a DOS program to be run, be it game or utility. It is sometimes required to FORCE the program to use what is termed RAW mode. This basically means that whenever the user presses a key while inside of the DOS program, the program responds as if the character was HOTKEYED. Normal dos-programs buffer characters up and do not respond back "as soon as a key is pressed", so for this reason the more professional doors require RAW to be either set, or do it themselves.

See the documentation of the specific DOS program you are wishing to run for more information.

1.61 ECHO

TITLE:

Echo on

TEMPLATE:

%ECHO.

FUNCTION:

DOS programs sometimes require the ECHO command to be given, which echoes typed characters onto the screen rather than have them hidden.

See the instructions for the particular DOS program you are wishing to run for more information on its calling sequence.

1.62 CRLF

TITLE:

Turn on both hard and soft RETURNS.

TEMPLATE:

%CRLF.

FUNCTION:

Some DOS programs require that CRLF be turned on. This command is the argument that does just that.

See the documentation of the specific DOS program you are wishing to run for more information.

1.63 CTRL_C

TITLE:

Control C (^C)

TEMPLATE:

%CTRL_C.

FUNCTION:

DOS programs generally have the ability to be **BREAKED** by pressing the Control C (^C) keys. This combination generally terminates the the program and control is resumed to the calling sequence. However, in most instances the ^C press should be turned off when dealing with DOS programs. This is for security reasons, as it can sometimes lead to users (via the DOS program) gaining accidental access to a full blown DOS-Shell. Unless "required" this command is not normally required to be specified when calling DOS programs.

If the command is ommited then ApT-BBS will ignore all presses of the ^C key, resulting in the DOS program never hearing the cries of the ^C key.

See the documentation of the specific DOS program you are wishing to run for more information.

1.64 CTRL_D

TITLE:

Control D (^D)

TEMPLATE:

%CTRL_D.

FUNCTION:

DOS programs can generally be terminated by a combintion of key presses. The Control C (^C) press is generally a **HARD** and **HARSH** way of terminating a program, and not recommended. However, to allow the "program" to know that a request has been made to "terminate cleanly" the ^D key can be used (providing that the program monitors for it) - This will result in the program exiting back to ApT-BBS without any problems.

It is a non-HARSH way of doing things, and always recommended that if the program accepts such a command, that it be supplied. Otherwise monitoring ofthe key by ApT-BBS will not be done, and the DOS program being used will never hear its cries.

See the documentation of the specific DOS program you are wishing to run for more information.

1.65 HANDLER

TITLE:

Handler Return

TEMPLATE:

```
%HANDLER.
```

FUNCTION:

When calling a DOS program there are various methods open to you. Sometimes "total" control over the method of calling is required, and for this reason this command is given. It allows certain programs that need to be run through "newshell" to be done by passing the correct handler to the argument list. This is quite technical and example files (both rexx and otherwise) are included within the package to give examples of how one might use this argument.

See the documentation of the specific DOS program you are wishing to run for more information.

EXAMPLE:

Within ApT-Rexx one might wish to Run the game called Hack&Slash,

```
/* Hack&Slash.ap
**
** $VER: Hack&Slash.ap 0.0.1 (25.4.93)
**
** ARexx program for running Hack&Slash Cli door by ApT-Design.
**
** Copyright © 1993 ApT-Design All Rights Reserved,AG
**
*/

parse arg lineno

options results

portname = 'APTREXX'lineno
address value portname

'GETSTR' USERNAME_STR ; name=result
gamedoor= 'DOORS:Hack2/HSCClient "'name'"'
path='t:door-startup'||lineno
open('temp',path,'W')
writeln('temp','stack 10000') ; writeln('temp', gamedoor )
writeln('temp','endcli') ; close('temp')
'RUNDOS' "run >nil: newshell >nil: %HANDLER. from "path" %RAW. %CRLF."
EXIT
```

With understanding of this complex calling sequence, it is possible to create a "virtual" Dos-Shell.

1.66 COL

TITLE:

Colour Flag

TEMPLATE:

%COL.

FUNCTION:

DOS programs sometimes require information about the users terminal preferences. Generally the request is to find out if the user is using a COLOUR display or a MONO display. This command returns the correct string pertaining to the preferences of the user.

See the documentation of the specific DOS program you are wishing to run for more information.
