

Keywords

BRAIN's complete keyword list.

See understanding [Keyword Syntax](#) for a description of the syntax BRAIN uses as well as some useful tips.

- ASK
- ALIAS
- AUTO SPEED
- AUTONOMOUS
- CLEAR ERRORS
- CLS
- DIGITALIO
- DISPOSE
- EXTRACT
- GLOBAL
- GOTO
- IF THEN ELSE
- LABEL
- MOTION DETECTION
- MEDIA PLAYER CONTROLS
- MOVE FORWARD
- MOVE REVERSE
- MOVE LEFT
- MOVE RIGHT
- ONERROR
- ONSUCCESS
- PAUSE SCRIPT
- PAUSE ALL SCRIPTS
- PLAY AUDIO
- PLAY VIDEO
- READ
- READ FILE
- RESTART SCRIPT
- RETURN
- ROTATE CW
- ROTATE CCW
- RUN COMMAND
- RUN SCRIPT
- RUN PROCESS
- SAFETY

Keywords

- SAY
- SHOW IMAGE
- SHOW IMAGE FROM URL
- SLEEP
- SPEED
- SPEED MULTIPLE
- STOP
- STRIP
- STOP SCRIPT
- STOP ALL SCRIPTS
- SPEECH SYNTHESIS
- SPEECH FOR SYSTEM MESSAGES
- SPEECH FOR NAVIGATION MESSAGES
- SPEECH RECOGNITION
- WAITKEY
- WARNINGS
- WRITE
- WRITE FILE

ASK

Read content from the Robot command console. Result is stored in a variable for later use in a script. Useful for prompting the user for some specific text.

Equivalent to the READ keyword.

Syntax:

`$variable=ASK (Question to ask)`

Example/s:

`ASK $name=What is your name?`

`SAY Thanks $name`

ALIAS

Assign a command sequence to a variable. The variable can later be called to execute the sequence of commands.

Keywords

Syntax:

ALIAS \$variable=command

ALIAS \$variable=command1;command2;command3; ... commandn

Example/s:

ALIAS \$blinkLights=DIGITALIO 16,TRUE;SLEEP 1000;DIGITALIO
16,FALSE

Say I am now going to blink the light.

\$blinkLights

AUTO SPEED

Toggle the automatic wheel speed setting.

Note: This value can also be adjusted via a user interface control in the wheel speed panel.

Syntax:

AUTO SPEED (ON|OFF)

Example/s:

AUTO SPEED ON

AUTONOMOUS

Toggle autonomous mode.

Note: This value can also be adjusted via a user interface control in the modes panel.

Syntax:

AUTONOMOUS (ON|OFF)

Example/s:

AUTONOMOUS ON

CLEAR ERRORS

Clear the current state of script errors.

For EXTRACT and READ FILE commands it is possible to receive an error (web site not available, file not found, etc). While an error notification may be desirable the first time, it is possible to suppress future notifications of the same error (via the error dialog box that is displayed). However, should you wish to turn this notification back on, you will need to clear the error state with the CLEAR ERRORS command. Typically this is placed at the start of a script before the EXTRACT or READ FILE commands.

Syntax:

CLEAR ERRORS

Example/s:

CLEAR ERRORS

CLS

CLear Screen – Clear the Robot command console.

Syntax:

CLS

Example/s:

The following script increments numbers one at a time, displaying the in the Robot command console, clearing the console in between each increment.

```
$count=0  
LABEL loop
```

Keywords

```
CLS  
$count=$count+1  
WRITE $count  
SLEEP 1000  
GOTO LOOP
```

DIGITALIO

Turn on or off a digital I/O port.
ON OFF, TRUE FALSE, 0 1, YES NO are all valid boolean values.

Syntax:

DIGITALIO (Port, ON|OFF)

Example/s:

DIGITALIO 7,ON

or

DIGITALIO 16,FALSE

DISPOSE

Dispose of a script variable so it no longer exists.

Syntax:

DISPOSE (\$variableName)

Example:

```
$count=5  
SAY $count
```

```
DISPOSE $count
```

```
SAY $count # This line would now cause a variable not defined  
error.
```

EXTRACT

Extract content from a web page and store the result in a variable. Generally best for XML content or very structured web sites. This command essentially parses out text between two text delimiters.

Syntax: (There are 4 variations of this command).

`$variable=EXTRACT (URL,LHS,RHS)`

Extract content from a web page and store the result in the variable `$variable`. If more than one pattern is found, extract only the first one. This command is the same as `EXTRACT FIRST`.

`$variable=EXTRACT ALL (URL,LHS,RHS)`

Extract content from a web page and store the result in the variable `$variable`.

`$variable=EXTRACT ANY (URL,LHS,RHS)`

Extract content from a web page and store the result in the variable `$variable`. If more than one pattern is found, extract one at random.

`$variable=EXTRACT LAST (URL,LHS,RHS)`

Extract content from a web page and store the result in the variable `$variable`. If more than one pattern is found, extract only the last one.

URL is the fully qualified URL of the web page (ex.

`http://www.website.com/file.html`

LHS (Left Hand Side) is the HTML text that precedes the text you wish to extract.

RHS (Right Hand Side) is the HTML text that comes after the text you wish to extract.

Example:

`$stockValue=EXTRACT`

`http://www.financesite.com/stockvalue.html,,`

SAY The Stock price is \$stockValue.

GLOBAL

Define a global variable. i.e. A variable that is available in all scripts.

Syntax:

GLOBAL (\$variable)

Example/s:

(Script 1).

```
GLOBAL $counter  
$counter=7
```

(Script 2).

```
Say $counter
```

GOTO

Jump to a pre-defined label in a script.

or

Jump to a pre-defined method block in the script.

Syntax: (There are three variations to this keyword).

GOTO (label)

Jump to a pre-defined label in a script.

GOTO LABEL (label)

Jump to a pre-defined label in a script. Same as GOTO.

GOTO METHOD (methodName)

Jump to a pre-defined method block in the script.

Keywords

Example/s:

```
$count=7  
IF $count=7 THEN GOTO odd Increment ELSE GOTO LABEL even
```

```
SAY $count is now $count
```

```
LABEL odd  
SAY $count is odd.  
GOTO done  
LABEL even  
SAY $count is EVENT  
GOTO done
```

```
LABEL done
```

```
$count=7  
GOTO METHOD Increment  
SAY $count is now $count
```

```
METHOD Increment  
$count=$count+1  
RETURN
```

IF THEN ELSE

Perform a logical IF THEN (with an optional ELSE) evaluation.

Syntax:

Keywords

IF (condition) THEN (command)
IF (condition) THEN (command) ELSE (optional command)

Example/s:

IF \$count=7 THEN SAY It's seven.

IF \$name=Robot THEN SAY Nice to meet you ELSE SAY Sorry, I don't know you.

IF \$OBJECT_DETECTED THEN GOTO warnUser

LABEL

Define a label in a script. Typically used with the GOTO keyword.

Syntax:

LABEL (labelName)

Example/s:

\$count=7

IF \$count=7 THEN GOTO odd Increment ELSE GOTO LABEL even

SAY \$count is now \$count

LABEL odd

SAY \$count is odd.

GOTO done

LABEL even

SAY \$count is EVENT

GOTO done

LABEL done

MOTION DETECTION

Toggle the state of the Robot's motion detection engine.

Note: This value can also be adjusted via a user interface control in

Keywords

the video panel.

Syntax:

MOTION DETECTION (ON|OFF)

Example/s:

SAY I am now turning on motion detection.
MOTION DETECTION ON

MEDIA PLAYER CONTROLS

Toggle the state of the media player controls. The media player panel is the panel that plays audio, video, and shows images. It can display controls that allow a user to play, pause, stop video and audio. Sometimes it is desirable to turn off (or on) these controls.

Syntax:

MEDIA PLAYER CONTROLS (ON|OFF)

Example/s:

MEDIA PLAYER CONTROLS OFF

MOVE FORWARD

Move the Robot forward "distance" units. Distance can be express as nMM, nCM, nI, nF to move the Robot n units in MM, CM, Inches, or Feet respectively.

Syntax:

MOVE FORWARD (distance)

Example/s:

MOVE FORWARD 9I
Moves the Robot forward 9 Inches.

Keywords

MOVE FORWARD 2f
Moves the Robot forward 2 feet.

MOVE REVERSE

Move the Robot in reverse "distance" units. Distance can be express as nMM, nCM, nI, nF to move the Robot n units in MM, CM, Inches, or Feet respectively.

Syntax:

MOVE REVERSE (distance)

Example/s:

MOVE REVERSE 9I
Moves the Robot in reverse 9 Inches.

MOVE REVERSE 2f
Moves the Robot in reverse 2 feet.

MOVE LEFT

Rotate the robot left "degrees" degrees.

Syntax:

MOVE LEFT (degrees)

Example/s:

MOVE LEFT 45

MOVE RIGHT

Rotate the robot right "degrees" degrees.

Keywords

Syntax:

MOVE RIGHT (degrees)

Example/s:

MOVE RIGHT 90

ONERROR

Mark the beginning and end of a script block to execute only if the previous command was unsuccessful. Used in conjunction with the ONERROR END keyword.

Applies only to the EXTRACT command.

Syntax:

ONERROR BEGIN
ONERROR END

Example/s:

\$stockValue=EXTRACT
<http://www.financesite.com/stockvalue.html>,

ONERROR BEGIN

Called if the extract failed. (For example, web site not available).
Say Extract failed.

ONERROR END

ONSUCCESS BEGIN

Called if the extract was successful.
SAY The Stock price is \$stockValue.

ONSUCCESS END

ONSUCCESS

Mark the beginning and end of a script block to execute only if the

Keywords

previous command was successful. Used in conjunction with the ONSUCCESS END keyword.

Applies only to the EXTRACT command.

Syntax:

```
ONSUCCESS BEGIN  
ONSUCCESS END
```

Example/s:

```
$stockValue=EXTRACT  
http://www.financesite.com/stockvalue.html,,
```

```
ONERROR BEGIN
```

```
# Called if the extract failed. (For example, web site not available).  
Say Extract failed.
```

```
ONERROR END
```

```
ONSUCCESS BEGIN
```

```
# Called if the extract was successful.  
SAY The Stock price is $stockValue.
```

```
ONSUCCESS END
```

PAUSE SCRIPT

Pause a specific script. (Script must be running).

Note: This value can also be adjusted via a user interface control in the status bar when scripts are active.

Syntax:

```
PAUSE SCRIPT (scriptName)
```

Example/s:

```
SAY I am now going to pause the talking clock script  
PAUSE SCRIPT TalkingClock
```

Keywords

PAUSE ALL SCRIPTS

Pause all currently running scripts.

Note: This value can also be adjusted via a user interface control in the status bar when scripts are active.

Syntax:

PAUSE ALL SCRIPTS

Example/s:

SAY I am now going to pause all scripts
PAUSE ALL SCRIPTS

PLAY AUDIO

Play an audio file in the media player.

Syntax:

PLAY AUDIO (Audio file)

Example/s:

PLAY AUDIO C:\music.wma

PLAY AUDIO \$CONTENT_DIRECTORY\song.wma

PLAY VIDEO

Play a video file in the media player.

Syntax:

PLAY VIDEO (Video file)

Example/s:

PLAY VIDEO C:\movie.wmv

PLAY VIDEO \$CONTENT_DIRECTORY\movie.wmv

READ

Read content from the Robot command console. Result is stored in a variable for later use in a script. Useful for prompting the user for some specific text.

Equivalent to the ASK keyword.

Syntax:

\$variable=READ (Question to ask)

Example/s:

READ \$name=What is your name?
SAY Thanks \$name

READ FILE

Read a text file from disk.

Syntax:

\$variable=READ FILE (Filename)

Example/s:

\$fileContents=READ FILE C:\foo.txt

\$fileContents=READ FILE \$CONTENT_DIRECTORY\foo.txt

SAY The file contains \$fileContents

See Also:

RESTART SCRIPT

Restart a script that has been paused.

Syntax:

RESTART SCRIPT (ScriptName)

Example/s:

RESTART SCRIPT TalkingClock

RETURN

Return from a METHOD call

Syntax:

RETURN

Example/s:

```
$count=7  
GOTO METHOD Increment  
SAY $count is now $count
```

```
METHOD Increment  
$count=$count+1  
RETURN
```

ROTATE CW

Rotate the Robot "degrees" degrees clockwise.

Syntax:

ROTATE CW (degrees)

Example/s:

Keywords

ROTATE CW 90

ROTATE CCW

Rotate the Robot "degrees" degrees counter-clockwise.

Syntax:

ROTATE CWW (degrees)

Example/s:

ROTATE CCW 90

RUN COMMAND

Execute an existing question.

Syntax:

RUN COMMAND (command)

Example/s:

Assumes a command called "Play Video" already exists.

SAY I will now play a video.

RUN COMMAND Play Video

RUN SCRIPT

Load and execute an existing script.

Syntax:

RUN SCRIPT (script)

Example/s:

RUN SCRIPT \$SCRIPT_DIRECTORY\PlayVideo

RUN PROCESS

Keywords

Run execute a Windows program (process).

Syntax:

RUN PROCESS (Complete path to program to run)

Example/s:

```
# Run the Windows calculator application.  
RUN PROCESS calc.exe
```

SAFETY

Toggle the Robot safety setting. (Turning safety off could cause the Robot to run into objects. Use with caution.)

Note: This value can also be adjusted via a user interface control in the modes panel.

Syntax:

SAFETY (ON|OFF)

Example/s:

SAFETY ON

SAY

Cause the Robot to audibly say specific text.

Syntax:

SAY (Text to say)

Example/s:

SAY Nice to meet you.

SAY The time is \$TIME

SHOW IMAGE

Load an image from the file system, and display it in the media player.

Syntax:

SHOW IMAGE (fullPathToImage)

Example/s:

```
SHOWIMAGE $CONTENT_DIRECTORY\$MyImage.jpg  
SHOW IMAGE C:\MyImage.png
```

SHOW IMAGE FROM URL

Obtain an image from a URL and display it in a the media player.

Syntax:

SHOW IMAGE FROM URL (Fully qualified URL to image)

Example/s:

```
SHOW IMAGE FROM URL http://www.site.com/image.jpg
```

SLEEP

Cause a script to pause for "time" mili-seconds. This essentially pauses the Robot unless other scripts are running in the background.

Syntax:

Keywords

SLEEP (Time in ms)

Example/s:

```
# A simple talking clock  
LABEL loop  
SAY The time is $TIME  
SLEEP 1000 # Sleep for 1 second (1000 milliseconds).  
GOTO loop
```

SPEED

Set the speed for both wheels.

Note: This value can also be adjusted via a user interface control in the wheel speed panel.

Syntax:

SPEED (speed)
Where speed is a value from 90–100.

Example/s:

```
SPEED 50 # Move at 1/2 speed.
```

SPEED MULTIPLE

Set the speed multiplier value. The Robot is capable of going quite fast. The speed multiple acts as a speed limiter for safety. Used in conjunction with the SPEED keyword, it controls the speed at which the Robot moves. The speed multiple multiplies the speed setting, so for example, a speed multiple of 2 enables the Robot to move twice as fast when adjusting the wheel speed (or speed command) settings.

Note: This value can also be adjusted via a user interface control in the settings options.

Syntax:

SPEED MULTIPLE (value)

Where (value) is a value from 1-4.

Example/s:

SPEED MULTIPLE 2

STOP

Cause the Robot to stop moving.

Note: This value can also be adjusted via a user interface control in the navigation controls panel.

Syntax:

STOP

Example/s:

STOP

STRIP

Remove extra whitespace and newlines from a variable. Often when reading a file or obtaining information from a web site, etc., the result may contain undesirable whitespace (spaces, tabs, etc), as well as newlines (empty lines between text lines). STRIP removes these.

\$variable=STRIP \$

\$fileContents=READ FILE \$CONTENT_DIRECTORY\foo.txt

SAY The file contains \$fileContents

STOP SCRIPT

Stop a currently executing script.

Note: This value can also be adjusted via a user interface control in the status bar when scripts are active.

Syntax:

STOP SCRIPT (ScriptName)

Example/s:

STOP SCRIPT Talking Clock

STOP ALL SCRIPTS

Stop all currently executing scripts.

Note: This value can also be adjusted via a user interface control in the status bar when scripts are active.

Syntax:

STOP ALL SCRIPTS

Example/s:

STOP ALL SCRIPT

SPEECH SYNTHESIS

Toggle the Robot's text to speech capability.

Note: This value can also be adjusted via a user interface control in the speech settings.

Syntax:

SPEECH SYNTHESIS (ON|OFF)

Example/s:

SPEECH SYNTHESIS ON

SPEECH FOR SYSTEM MESSAGES

Toggle whether or not the Robot uses speech to report system events. (Ex saying things such as "Starting Robot Services.").

Note: This value can also be adjusted via a user interface control in the speech settings.

Syntax:

SPEECH FOR SYSTEM MESSAGES (ON|OFF)

Example/s:

SPEECH FOR SYSTEM MESSAGES OFF

SPEECH FOR NAVIGATION MESSAGES

Toggle whether or not the Robot uses speech to report navigation events. (Ex Saying things such as "Moving Forward 10 inches.").

Note: This value can also be adjusted via a user interface control in the speech settings.

Syntax:

SPEECH FOR NAVIGATION MESSAGES (ON|OFF)

Example/s:

SPEECH FOR NAVIGATION MESSAGES OFF

SPEECH RECOGNITION

Keywords

Toggle the Robot's speech recognition capability.

Note: This value can also be adjusted via a user interface control in the speech settings.

Syntax:

SPEECH RECOGNITION (ON|OFF)

Example/s:

SPEECH RECOGNITION OFF
Say I can't hear you any more.

WAITKEY

This command will pause the script from running until a key is pressed.

Syntax:

WAITKEY

Example/s:

SAY Press and key to continue...
WAITKEY
Say I am now continuing.

WARNINGS

Toggle script warnings. If OFF, script errors will not be displayed.

Syntax:

WARNINGS (ON|OFF)

Example/s:

SAY I will not complain about script errors.
WARNINGS OFF

WRITE

Write a message to the Robot's command panel.

Syntax:
WRITE (Message)

Example/s:

```
$count=7  
GOTO METHOD Increment  
WRITE $count is now $count
```

```
METHOD Increment  
$count=$count+1  
RETURN
```

WRITE FILE

Write content to (and optionally create) a text file.

Syntax:

WRITE FILE (Filename,TRUE|FALSE,content)

Filename is the name of the file to write to (or create).
TRUE|FALSE specifies whether or not to allow automatic overwriting
of existing files.
Content is the data you wish to write to the file.

Example/s:

```
# Writes "This is a line of text." to the file C:Log.txt. If the file  
exists, it will be overwritten.  
WRITE FILE C:\Log.txt,TRUE,This is a line of text.
```

Keywords

```
# Same example, except create the file in BRAIN's content  
directory.  
WRITE FILE $CONTENT_DIRECTORY\Log.txt,TRUE,This is a line of  
text.
```
