

# The SCUMM Manual

## Glossary

<b>abs</b>	ref func
var=abs expression	
This function calculates the absolute value of the given parameter.	
<b>actor</b>	ref comm
actor actor-name	
This command is used to initialize and change the facets of an actor. Multiple arguments can be combined into a single statement.	
<b>actor-box</b>	ref func
var=actor-box actor-name	
This function returns the walk box of the specified actor.	
<b>actor-chore</b>	ref func
var=actor-chore actor-name	
This function returns the last choreography executed by the actor. Includes walks and talks.	
<b>actor-costume</b>	ref func
var=actor-costume actor-name	
This function returns the costume of the specified actor.	
<b>actor-depth</b>	ref func
var=actor-depth actor-name	
This function returns the depth of the specified actor.	
<b>actor-elevation</b>	ref func
var=actor-elevation actor-name	
This function returns the elevation of the specified actor.	
<b>actor-facing</b>	ref func
var=actor-facing actor-name	
This function returns the directional facing of the specified actor. This value is an angle with NORTH at 0, EAST at 90 etc.	
<b>actor-inventory</b>	ref func
count=actor-inventory actor-name	
This function returns the number of items in the specified actor's inventory as well as the items.	
<b>actor-moving</b>	ref func
var=actor-moving actor-name	
This function returns whether the actor is stopped or moving.	
<b>actor-room</b>	ref func
var=actor-room actor-name	
This function returns the room number of the specified actor.	

<b>actor-scale</b>	ref comm
var=actor-scale actor-name	
This function returns the scale of the specified actor.	
<b>actor-scale-array-string</b>	svar
System variable. Allows users to control the actor scale table.	
<b>actor-talking</b>	svar
System variable hold the last actor who spoke.	
<b>actor-width</b>	ref func
var=actor-width actor-name	
This function returns the width of the the specified actor.	
<b>actor-x</b>	ref func
var=actor-x actor-name	
This function returns the x room coord of the specified actor.	
<b>actor-y</b>	ref func
var=actor-y actor-name	
This function returns the y room coord of the specified actor.	
<b>actor-zplane</b>	gloss ref
var=actor-zplane actor-name	
This function returns the zplane of the specified actor.	
<b>actor.def</b>	file
Assigns a number to each actor. Global chores are also defined here. Created by MAKEDEFS.	
<b>.anm</b>	file
Files generated by Deluxe Paint Animator.	
<b>array</b>	ref lang
compile time (static) array:	
Arrays can be one or two dimensional. Arrays can be static or dynamic.	
<b>array-shuffle</b>	ref comm
array-shuffle foo[10] to foo[20]	
This command randomizes elements in a one dimensional array	
<b>bit variable</b>	ref lang
bit variable variable-name	
A variable with a value of 1 or 0, true or false.	
<b>blast-object</b>	ref comm
blast-object object at x-pos , y-pos	
A very fast drawing routine that can draw lots of things to the screen at any x,y SCREEN coordinate.	
<b>blast-text</b>	ref comm
blast-text "string" at x-pos , y-pos [charset font color col wrap]	
A very fast text drawing routine that can display text at any x,y position.	
<b>both-buttons-clicked-key</b>	svar
System variable. Key passed if both controller buttons pressed.	

<b>break-here</b>	ref flow
break-here [number]	
Simple break away command. Essential to our method of multitasking. Can be used to time animation sequences.	
<b>break-until</b>	ref flow
break-until (condition-met)	
A combination of a break-here and a do-until.	
<b>break-while</b>	ref flow
break-while (condition-met)	
breaks will occur if condition met	
<b>build-sentence-script</b>	svar
System variable. Script to run when button hit.	
<b>camera</b>	ref comm
camera pause	
Stop the camera from moving	
<b>camera-accel-x/y</b>	svar
System variables. How quickly the camera accelerates.	
<b>camera-at</b>	ref comm
camera-at x-coord [y-coord]	
This command sets the camera position in a room. The values indicates the midpoint of the screen.	
<b>camera-follow</b>	ref comm
camera-follow actor-name	
This command tells the camera to follow the specific actor around the room.	
<b>camera-follow-actor</b>	svar
System variables which holds the current actor being followed.	
<b>camera-max-x/y</b>	svar
System variables which holds the maximum position for the camera.	
<b>camera-min-x/y</b>	svar
System variables which hold the minimum position for the camera.	
<b>camera-pan-start-x/ydist</b>	svar
System variables. How far from actor before pan starts.	
<b>camera-pan-to</b>	ref comm
camera-pan-to x-coord [y-coord]	
This command tells the camera to smoothly pan to a new position in the room.	
<b>camera-script</b>	svar
System variable. Script to run when camera moves out of sync with other scripts. Used for parallax effects.	
<b>camera-speed-x/y</b>	svar
System variables. Maximum speed for the camera.	

<b>camera-x</b>	svar
System variable which holds camera x position in pixels. 320 is center screen.	
<b>camera-y</b>	svar
System variable which holds camera y position in pixels. 200 is center screen.	
<b>case</b>	ref flow
<pre>case case-name {</pre> Multiple choice branching statement.	
<b>chain-script</b>	ref flow
<pre>chain-script [bak rec] script-name [(] [^1[,]^2] [)]</pre> This command loads and executes a new script and stops the calling script.	
<b>charset*</b>	ref lang
<pre>charset [charset-name]</pre> The charset command sets the current character set being used. The system will load the charset onto the heap if it is not already there. *UNDER RECONSTRUCTION	
<b>chore</b>	term
A choreography.	
<b>class-of</b>	ref func comm
<pre>class-of object is object-class</pre> This command is used to set the classification of objects and actors. It can also be used as a function to check the classification.	
<b>#code on   off</b>	ref lang
<pre>#code on   off</pre> Placing these in you code will cause the compiler to output source and object code to the screen. Used for debugging and the development of the language.	
<b>come-out-door</b>	ref flow
<pre>come-out-door object-name \</pre> This command is used to change rooms. Places the selected-actor in a new room.	
<b>#comment</b>	ref lang
<pre>#comment "string"</pre> Allows the script to display messages during compile time	
<b>controller</b>	svar
System variable. Type of game controller.	
<b>.cos</b>	file
Run length, compacted versions of .cst files. The system requires that animations be in .cos format.	
<b>costumes.def</b>	file
Every costume must be defined in this file. You can force the system to create this file by running MAKEDEFS.	
<b>.cst</b>	file
Files generated by CYST.	

<b>current-room</b>	ref flow
current-room room-name	
This command changes the camera to a new room. In the process it causes the exit code from the last current room to be run and then the enter code in the new current room.	
<b>current-year/month/day/hours/minutes/seconds</b>	svar
System variable for the user's computer time. Must call get-time-date before these are read.	
<b>cursor</b>	ref comm
cursor object-number [image flames-state]	
This command sets the shape of the cursor.	
<b>cursor hotspot</b>	ref comm
cursor hotspot x y	
This command sets the hotspot of the cursor.	
<b>cursor onloff</b>	ref comm
cursor on off soft-on soft-off	
This command allows you to remove and disable the cursor during cutscenes.	
<b>cursor transparent</b>	ref comm
cursor transparent color	
This command sets transparent colors in the cursor. It can be called multiple times for multiple transparent colors.	
<b>cursor-room-x</b>	svar
System variable for the x position of the cursor in the room.	
<b>cursor-room-y</b>	svar
System variable for the y position of the cursor in the room	
<b>cursor-screen-x</b>	svar
System variable. Gives the location of the cursor in SCREEN coords	
<b>cursor-screen-y</b>	svar
System variable. Gives the location of the cursor in SCREEN coords	
<b>cursor-state</b>	svar
System variable.	
<b>cut-scene</b>	ref flow
cut-scene [type]{	
A cut-scene is a computer controlled sequence. During a cut-scene, the player has no control of the game. All other scripts, except BAK scripts, are temporarily halted during the cut-scene.	
<b>cut-scene1-script</b>	svar
System variable. Script to run on start of cut-scene.	
<b>cut-scene2-script</b>	svar
System variable. Script to run on end of cut-scene.	
<b>CYST</b>	app
The animation tool used primarily for walk talk animations. It breaks an animation down into limbs that can then be manipulated independently.	

<b>debug</b>	ref comm
debug debug-level [windex command]	
Used to start windex or execute windex commands from scripts.	
<b>define</b>	ref lang
define symbol	
Used to define any symbol.	
<b>define-local-script</b>	ref lang
define-local-script local-script-name	
Local scripts do not usually need to be defined. However, in special situations, they must be defined using define-local-script. This allows you to mix auto-defining-scripts with predefined scripts.	
<b>define-script</b>	ref lang
define-script script-name	
Used to define scripts.	
<b>delcos.bat</b>	app
Created by makedefs. It contains all unused .cos files in your project's directory tree. Running this batch file will delete all these unused files. USE WITH CAUTION.	
<b>dellbm.bat</b>	app
Created by makedefs. It contains all unused .lbm files in your project's directory tree. Running this batch file will delete all these unused files. USE WITH CAUTION.	
<b>delscu.bat</b>	app
Created by makedefs. It contains all unused .scu files in your project's directory tree. Running this batch file will delete all these unused files. USE WITH CAUTION.	
<b>Deluxe Paint</b>	app
The art program used for background art. Made by Electronic Arts.	
<b>Deluxe Paint Animator</b>	app
The animation program used for 2D character animation. Made by Electronic Arts.	
<b>dependent-on (retired)</b>	ref retired
dependent-on object-name being object-state	
This command makes the state of one object dependent on the state of another object.	
<b>dim</b>	ref comm
dim [local] [string]array variable [size]	
This command dynamically allocates an array and stores a pointer to the array.	
<b>display-version-key</b>	svar
System variable. Which key causes version displays to appear.	
<b>DK</b>	app
The very powerful art tool for color reduction and manipulation. Most of the manipulation of palettes in finished art work is done in DK.	
<b>do [until]</b>	ref flow
do {	
The do construct is used to repeat a section of code. The do until command is used to repeat a section of code until a condition is satisfied.	

<b>do-animation</b>	ref comm
do-animation actor-name choreography	
This command tells an actor's costume which animation sequence to perform.	
<b>do-animation face-towards</b>	ref comm
do-animation actor-name face-towards actor-name	
This command causes an actor to face towards another actor.	
<b>do-sentence</b>	ref flow
do-sentence verb-name object-name	
This command lets the programmer artificially construct a sentence which will execute as if the player had controlled it.	
<b>Dpaint</b>	app
see deluxe paint.	
<b>draw-box</b>	ref comm
draw-box x-coor,y-coor to x-coor,y-coor color color-name	
This command paints a solid colored box on the screen.	
<b>draw-object</b>	ref comm
draw-object object-name [at x-coord,y-coord]	
This command draws an object to the screen.	
<b>#else</b>	ref lang
#else	
A compile else follows a #if. It is checked at compile time.	
<b>enter-room1-script</b>	svar
System variable. Script to run BEFORE enter code.	
<b>enter-room2-script</b>	svar
System variable. Script to run AFTER enter code.	
<b>entered-door</b>	svar
System variable indicate object that actor came out of.	
<b>environment variable</b>	term
Set in scumm.ini file, these variables set the paths for the project. For example: ARTPATH=c:\indy5\ART\ causes all utilities using ARTPATH to search in this directory. Typing ART at the dos prompt will take the scriptor to the ART directory.	
<b>#error</b>	ref lang
#error "string"	
Allows the script to display messages during compile time	
Stops compilation	
<b>exit-room1-script</b>	svar
System variable. Script to run BEFORE exit code.	
<b>exit-room2-script</b>	svar
System variable. Script to run AFTER exit code.	
<b>facing*</b>	ref func
var=facing actor-name	
This function returns the actor or object's current facing angle.	
Angles are 0-360 degrees with 0 being North, 90 being East	

<b>fade-delay</b>	svar
System variable. Number of piffies (1/240 th of a second)between fade steps. Default is 3.	
<b>fades*</b>	ref comm
fades fade-name	
This command controls the screen transition during the next room transition. *nonfunctional in 640x480 games	
<b>find-actor</b>	ref func
var=find-actor x-coord, y-coord	
This function finds the front-most actor at the given SCREEN coordinates.	
<b>find-all-objects</b>	ref func
foo = find-all-objects room-name	
This function returns an array listing all of the objects in the current room.	
<b>find-inventory*</b>	ref func
var=find-inventory actor-name, slot	
This function returns the inventory object in the given slot of the specified actor.	
<b>find-object</b>	ref func
var=find-object x-coord, y-coord	
This function returns the front-most object at the given ROOM coordinates. If no object is present at the x and y position, the value returned is a 0.	
<b>find-verb</b>	ref func
var = find-verb xcoord ycoord	
This function returns the verb at the given SCREEN coordinates.	
<b>Flem</b>	app
The room layout tool for objects and walk boxes.	
<b>for</b>	ref flow
for variable = value to value ++   -- {	
The variable is initialized with the first value. The system continues through the loop and on each pass, the variable is incremented (++) or decremented (--) until it equals the last value. It is possible to have for loops with sets.	
<b>frame-jiffies</b>	svar
System variable. Real jiffies that the last frame took to me displayed.	
<b>freeze-scripts</b>	ref flow
freeze-script	
This command is called by the cut-scene statement and freezes all scripts except BAK scripts. This command is used in the system level scripts, cutscenes,enter and exit scripts, dialogs.	
<b>game-loaded</b>	svar
if (game-loaded)	
System variable set if game loaded from the heap.	
<b>get-time-date</b>	ref comm
get-time-date	
This command causes the current-time variables to be set.	
<b>#if</b>	ref lang
#if expression	
A compile if is checked at compile time.	



<b>.lbn</b>	file
Background art files.	
<b>left-button-state</b>	svar
System variable. Scripts can get button position. 1 is down, 0 is up.	
<b>#lex on   off</b>	ref lang
#lex on   off	
Used for debugging. It takes text and converts it to tokens.	
<b>.lfl</b>	file
Contain compiled code for each room.	
<b>lflfile</b>	file
Created by MAKEMAKE. It is used to show dependencies used by MAKE to determine if a room must be recompiled.	
<b>load-</b>	ref comm
load-costume costume-name	
This command loads the specified item from a disk onto the heap.	
<b>load-game</b>	ref flow
load-game	
This command loads the saved game from the heap	
<b>load-lock-</b>	ref macro
load-lock-costume costume-name	
The load-lock macros loads items onto the heap and then locks them. Macros defined in scummmac.def.	
<b>local variable</b>	ref lang
local variable variable-name	
A variable that is local to a script.	
<b>lock-</b>	ref comm
lock-costume costume-name	
This command sets a flag which prevents an item from being removed from the heap once it's loaded in.	
<b>macro</b>	ref lang
macro macro-name ^1 ^2 {	
Allows scripters to create more complex commands	
<b>macros.def</b>	file
Frequently needed functions that have not yet been implemented into the system are written as macros in this file.	
<b>make.exe</b>	app
A DOS utility which is called by mk and mkall.	
<b>makedefs.exe</b>	app
Makes the objects.def, scripts.def, costumes.def and sounds.def files from the .scu files	
<b>makemake.exe</b>	app
Makes files (roofile and lflfile) that MK.BAT and MKALL.BAT use. It also builds the dependencies for .cos, .sou and INCLUDE files. It must be run every time a new room is added.	

<b>max-number-objects</b>	svar
System variable. Highest object # in the game.	
<b>memory-speed</b>	svar
System variable. Speed of memory copy.	
<b>message-going</b>	svar
System variable. 1 = message going.	
<b>min-jiffies</b>	svar
System variable governor for minimum number of jiffies per frame.	
<b>min-jiffies-per-char</b>	svar
System variable. Determines text delay on messages.	
<b>min-jiffies-per-message</b>	svar
System variable. Determines text delay on messages.	
<b>mk.bat</b>	app
The batch file that brings all files up to date, checks to see which files have been changed and calls MAKE. Use mk when changes are made to the .scu only.	
<b>mkall.bat</b>	app
The batch file that brings all files up to date, checks to see which files have been changed and calls MAKE. Use mkall when changes have been made to anything beyond the .scu file.	
<b>mmucus.exe</b>	app
The room compressor for SCUMM. It builds .roo files from .lbn and .ifo data.	
<b>name is</b>	ref comm
name is "object-name"	
This command is used at the beginning of every object definition to provide the name displayed when "%o" is used in a print command.	
<b>new-name-of</b>	ref comm
new-name-of object-name is "string"	
This command changes the "string" name of an object.	
<b>next-room</b>	svar
System variable. Only valid during EXIT code.	
<b>nuke -</b>	ref comm
nuke-costume costume-name	
This command sets the time stamp to be the oldest possible item on the heap. This will cause it to be thrown away first.	
<b>object-facing</b>	ref func
var = object-facing object-name	
This function returns the use-angle (0-359) of the specified object.	
<b>object-image-height</b>	ref func
var=object-image-height object-name	
This function returns the height in pixels of the object's image.	

**object-image-width** ref func

var=object-image-width object-name

This function returns the width in pixels of the object's image.

**object-image-x** ref func

var=object-image-x object-name

This function returns the x coordinate of the object's image in ROOM coords

**object-image-y** ref func

var=object-image-y object-name

This function returns the y coordinate of the object's image in ROOM coords

**object-running** ref func

var = object-running object-name

This function is used to tell if the specified object is currently running.

**object-x** ref func

var=object-x object-name

This function returns the value of the x-coordinate of the use-position of an object in ROOM coords.

**object-y** ref func

var=object-y object-name

This function returns the value of the y-coordinate of the use-position of an object in ROOM coords.

**objects.def** file

A list and definition of all the objects in the game. It is created by MAKEDEFS. The scriptor should never need to edit this file.

**override** ref flow

override label-name

Used to jump over long sections of a game, especially cutscenes when a special key (usually escape) is pressed by the player. This lets the player skip long dialogs or long non-interactive sections of the game.

**override-hit** svar

System variable. True if override hit.

**override-key** svar

System variable hold the key to be used for override.

**owner-of** ref func

owner-of object-name is actor-name

This command is used to assign ownership of an object to an actor. Can also be a function which returns a value of who is the owner of an object.

**palette** ref comm

palette red green blue in-slot slot-name

This command is used to manipulate colors and palettes.

**palette cycle-speed** ref comm

palette cycle-speed cycle-slot is speed

This command adjusts the color cycle speed of the specified cycle-slot. This is can also be used to turn color cycling off.

**palette intensity** ref comm

palette intensity intensity-value in-slot slot to slot

This command changes the intensity of a color or a group of colors in the palette. Can also be used to adjust the intensity level of the red, green and blue values independently.

**palette transform** ref comm

palette transform palette-number \

This command switches between alternate palettes.

<b>password-string</b>	svar
System variable.	
<b>pause-key</b>	svar
System variable. Key used for game pause.	
<b>pick</b>	ref func
var = pick expression of [expression...]	
This function picks a number from a list based on an index.	
<b>pick-up-object</b>	ref func
pick-up-object object-name	
This command picks up an object, puts it into the selected-actor's inventory (sets ownership), automatically sets the state of the object on the screen to GONE, and makes it UNTOUCHABLE.	
<b>pixel</b>	ref func
var= pixel x-screen-coord, y-screen-coord	
This function returns the value of the pixel at the given SCREEN coords	
<b>print-cursor</b>	ref comm
print-cursor [at x-coord, y-coord]	
This command prints text to the current cursor image	
<b>print-debug</b>	ref comm
print-debug [at x-coord, y-coord]	
This command prints a string to the windex debugging monitor.	
<b>print-line</b>	ref comm
print-line [left]	
This command prints a string to the screen. Used for messages from non-actors (a PA system or when there are no actors present.).	
<b>print-system</b>	ref comm
print-system [color color-number]	
This command prints a string to the message box (ala pause).	
<b>print-system-key-pressed</b>	svar
System variable set after calling 'print-system'	
<b>project.ifo</b>	file
Contains a list of defines, object templates, room templates and the list of the rooms in the project.	
<b>proximity</b>	ref func
var=proximity actor-name actor-name	
This function returns the distance between two actors/objects or two points.	
<b>put-actor</b>	ref comm
put-actor actor-name [at x-coord,y-coord]	
This command puts an actor at a specified location.	
<b>put-cursor</b>	ref comm
put-cursor x,y	
Places the cursor at the given screen coordinates	
<b>quit</b>	ref flow
quit	
This command exits the game.	

<b>random</b>	ref func
var=random number	
This function returns a random number from 0 to the number indicated	
<b>random-between</b>	ref func
var = random-between lower-number to upper-number	
This function returns a random number between two numbers. Random value is inclusive of the end values	
<b>restart</b>	ref flow
restart	
This command restarts the game from the beginning. It clears all variables, terminates all running scripts, empties the heap, and starts the boot-script over again.	
<b>restart-key</b>	svar
System variable. Key used for game restart.	
<b>return</b>	ref flow
return variable	
Allows a script to return a value.	
<b>right-button-state</b>	svar
System variable. Scripts can get button position. 1 is down and 0 is up.	
<b>.roo</b>	file
Contain compiled flem and art information.	
<b>roofile</b>	file
Created by MAKEMAKE, it is used to show dependencies used by MAKE to determine if a room must be recompiled.	
<b>room-height</b>	svar
System variable. Height for the current room.	
<b>room-width</b>	svar
System variable. Width of the current room (imagemin)	
<b>run-script</b>	ref macro
run-script [bak][rec] script [^1 ^2]	
This is a macro that starts a script and then breaks until the script is not running.	
<b>save-game</b>	ref flow
save-game	
This command saves the state of the game to the heap after the next frame.	
<b>save-load-colors</b>	svar
System variable. The colors for the control panel are in here.	
<b>save-load-enter-script</b>	svar
System variable. Script that gets run before each saveload.	
<b>save-load-exit-script</b>	svar
System variable. Script that gets run after each saveload.	

<b>save-load-key</b>	svar
System variable. Key used for saveload screen. 0 for no saveload.	
<b>saveload-enter-script</b>	svar
System variable. The script that gets run at start of saveload.	
<b>saveload-error</b>	svar
System variable. Saveload error flag for autosave and autoload.	
<b>say-line</b>	ref comm
say-line [actor-name] ["string"[[.][+]][:]	
This command prints a line to the screen and causes the speaker's mouth to animate.	
<b>sayline-override-key</b>	svar
System variable. Key for overriding say-lines.	
<b>script-running</b>	ref func
var = script-running script-name	
This function returns TRUE if the specified script is currently running.	
<b>script-version-string</b>	svar
System variable. Version of game to be displayed when 'display-version-key' hit.	
<b>scripts.def</b>	file
Every global script must be defined in this file. You can force the system to create this file by running MAKEDEFS.	
<b>.scu</b>	file
The SCUMM file. This file contains all of the code.	
<b>scumm.exe</b>	app
The compiler for the SCUMM language.	
<b>scumm.ini</b>	file
Sets the environment variables for the project. This makes it possible to have several project environments on one computer and simply switch between them. The environment variables set the paths for the project.	
<b>selected-actor</b>	svar
System variable used to hold who the current actor is.	
<b>selected-room</b>	svar
System variable which holds the current room number.	
<b>sentence-script</b>	svar
System variable. Script to run from do-sentence.	
<b>set-box</b>	ref comm
set-box box-number [box-number ...] to box-status	
This command allows the programmer to change the status of one or more boxes.	

<b>set-box-path</b>	ref comm
set-box-path	
This command rebuilds the box-connectivity table.	
<b>set-box-set</b>	ref comm
set-box-set set	
THIS COMMAND NEEDS DOCUMENTATION.	
<b>sets</b>	term
Groups of numbers that can be tested against. These can be represented as actors [guybrush,elaine,wally], objects [door, chair, window] or any series of numbers [1,3,5,7]	
<b>sleep-for</b>	ref flow
sleep-for number [jiffy jiffies]	
Causes a break measured in minutes, seconds or jiffies.	
<b>.sou</b>	file
Sound and music effect files.	
<b>sound</b>	ref comm
sound param [param...]	
A command hook for imuse commands. Do not call directly. Use macros in imuse.def.	
<b>sound-running</b>	ref func
var = sound-running sound-name	
This function returns TRUE if the specified sound is currently running.	
<b>special case animation</b>	term
Animations, usually choreographed in CYST, that allow an actor to do something other than walk and talk. These animations are usually only use once in a game.	
<b>sputm-debug</b>	svar
System variable. SPUTMDEBUG environment variable allows ^g jumps.	
<b>sputm.exe</b>	app
The interpreter for the SCUMM language.	
<b>stamp-actor</b>	ref comm
stamp-actor actor at x, y [scale scaleVal]	
This command draws an actor into the room.	
<b>stamp-object</b>	ref comm
THIS COMMAND NEEDS DOCUMENTATION.	
<b>start-music</b>	ref comm
start-music music-name	
This command begins playing a piece of music.	
<b>start-object</b>	ref flow
start-object [bak][rec] object-name verb verb-name	
This command launches the code within an object/verb immediately, rather than having the actor walk over to the object first as part of a do-sentence execution. It can also be use as a function.	
<b>start-script</b>	ref flow
[start-script] [bak][rec] script-name [() [^1 ^2] ()]	
This command tarts the execution of a new script, automatically loading it onto the heap if it's not already there. It can be run as a function.	

<b>start-sfx</b>	ref comm
start-sfx effect-name	
This command begins playing a sound effect	
<b>start-variables</b>	ref lang
start-variables [expression] [room-locals]	
Prefaces a list of variables that are local to a room. This tells the system how to start numbering the variables that follow.	
<b>start-video</b>	ref comm
start-video "video name"	
This command begins playing a video sequence	
<b>state-of</b>	ref comm func
state-of object-name is state	
This command sets the state of an object. Can also be used as a function.	
<b>stop-actor</b>	ref macro
stop-actor actor-name	
Used to stop an actor from walking. This is just a macro short-cut for 'actor <foo> stop' in scummmac.def	
<b>stop-object</b>	ref flow
stop-object object-name	
This command stops the object's verb code.	
<b>stop-script</b>	ref flow
stop-script [script-name]	
This command stops the execution of a script.	
<b>stop-sentence</b>	ref flow
stop-sentence	
This command terminates the sentence script and clears the do-sentence stack.	
<b>stop-sound</b>	ref comm
stop-sound sound-name	
This command stops a sound.	
<b>string-width</b>	ref func
var = string-width charset foo "string"	
This function returns the width in pixels of the specified string	
<b>system variable</b>	term
Variables used by the system. They are declared and commented in SCUMMSYS.DEF	
<b>undim</b>	ref comm
undim variable	
This command deallocates the space on the heap that was reserved for an array.	
<b>unfreeze-scripts</b>	ref flow
unfreeze-scripts	
This command unfreezes all scripts. It is called by the cut-scene statement system level scripts, enter and exit scripts and dialogs.	
<b>unlock-</b>	ref comm
unlock-costume costume-name	
This command unlocks the item on the heap but doesn't remove it.	
<b>update-inven-script</b>	svar
System variable. Script to update inventory.	

<b>userput</b>	ref comm
<p>userput on off soft-on soft-off</p> <p>This command activates and deactivates the keyboard and mouse.</p>	
<b>userput-state</b>	svar
<p>System variable.</p>	
<b>valid-verb</b>	ref func
<p>var=valid-verb object-name, verb-name</p> <p>This function returns TRUE if the verb listed is within the object's definition, otherwise it returns FALSE.</p>	
<b>variable</b>	ref lang
<p>variable variable-name [number] [number] [] [...]</p> <p>Declares a variable.</p>	
<b>variables.def</b>	file
<p>All global variables are declared here. All new variables must be added to the end of the appropriate list.</p>	
<b>verb</b>	ref comm
<p>verb verb-name [at x-coord,y-coord]</p> <p>The verb statement controls the look of the interface.</p> <p>HUH?</p>	
<b>verb-delete</b>	ref comm
<p>verb-delete verb-name</p> <p>This command kills the whole definition of the verb from the screen and the heap.</p>	
<b>verb-x</b>	ref func
<p>var = verb-x verb-name</p> <p>This function returns the x position of the specified verb in SCREEN coords</p>	
<b>verb-y</b>	ref func
<p>var = verb-y verb-name</p> <p>This function returns the y position of the specified verb in SCREEN coords</p>	
<b>verbs-delete</b>	ref comm
<p>verbs-delete from-verb to to-verb set verb-set</p> <p>THIS COMMAND NEEDS DOCUMENTATION.</p>	
<b>verbs-restore</b>	ref comm
<p>verbs-restore from-verb to to-verb set verb-set</p> <p>THIS COMMAND NEEDS DOCUMENTATION.</p>	
<b>verbs-save</b>	ref comm
<p>verbs-save from-verb to to-verb set verb-set</p> <p>THIS COMMAND NEEDS DOCUMENTATION.</p>	
<b>video-speed</b>	svar
<p>System variable. Speed of copies to video.</p>	
<b>wait-for-actor</b>	ref flow
<p>wait-for-actor actor-name</p> <p>This command waits before running the next line of code until the actor gets to its destination.</p>	
<b>wait-for-animation</b>	ref flow
<p>wait-for-animation actor-name</p> <p>This command waits before running the next line of code until the actor finishes an animation.</p>	

**wait-for-camera** ref flow

wait-for-camera

This command waits before processing the next line of code until the camera has stopped moving.

**wait-for-message** ref flow

wait-for-message

This command causes a break-until the current message has finished being displayed.

**wait-for-sentence** ref flow

wait-for-sentence

This command waits until all do-sentence's in the stack have been executed.

**wait-for-turn** ref flow

wait-for-turn actor-name

This command waits before running the next line of code until the actor finishes turning.

**walk** ref comm

walk actor-name to [x-coord,y-coord]

This command instructs an actor to walk to a specific location in the room. This will turn the actor to face the direction of the walk destination and then walk the actor to that location.

**walk talk animation** term

Animations, usually done in CYST, that allow an actor to walk and talk.

**while** ref flow

while (condition) {

While is similar to until, but the condition test is performed BEFORE any statements are executed.

**.zb?** file

Zplane clipping files.