

CAOS LANGUAGE GUIDE

Introduction

This document describes the commands available in the CAOS language, the language with which you can change, influence and create the inhabitants of Albia.

Teaching the art of programming is beyond the scope of this text so it mainly deals with the type of commands available and their usage.

With the CAOS language – and a tool to inject CAOS into Albia - you can query the condition of any item, create new ways for an object to work or make a whole new object to distribute around the world for all to use.

Before detailing the actual commands we'll start with a breakdown of what makes up an object in Albia. Throughout this document it is assumed that you have some means of injecting CAOS into Creatures and a means of getting results from injected commands.

So what is an Object?

An object (or agent – the two terms will be used interchangeably throughout this document) is an event driven entity within the world of Albia. These can include things as diverse as beach-balls, plants, animals, machinery, medicines – in fact anything that isn't a living Creature.

Being event driven means that the agent responds to events, and in Albia events can be activated by a creature, received a signal from another agent, hit the floor, or some other defined event.

A complete object then will consist of two main 'parts' which define how it is, acts, moves etc:

- Creation and initialisation - This will consist of the commands necessary to create the in-game presence of the object, define it's physical attributes and the kind of events it responds to, move it into the world and start it up. The creation of the object will also be where it is assigned a unique classifier – it is this classifier that marks the object and it's event scripts.

- Events - These are the events that an object will respond to and define how it reacts. Event code is stored in a special place within the world called the *scriptorium.*

And what is an Event?

An event is a chunk of script that an object runs, the process of triggering an event can arise from different circumstances. Some events come about because of the world – collision events, for example, are triggered whenever an agent that responds to boundaries hits a surface that it can not pass through.

Other events can arise from user/creature interaction – pickup and drop events, for example, occur whenever the object is picked up or dropped. Other events can be defined so that they occur at a user defined time – an activate 1 event, for example, usually occurs from a click on the object, and timer events can be used to allow periodic running of a script.

And what is Targ?

All CAOS operators work on the current targ object and the assumption is made that the commands that are processed are all to do with this targ. Targ can be changed using a few commands but the default targ is always the owner of the event script which is being processed, that is to say if a particular object's collision script is being processed then it will be targ until targ is programmatically changed. One way that targ changes is by using a NEW: command to create a new object of a certain type – for all the commands after the NEW: targ is referring to the new object just created and not the owner of the script. Other ways of changing targ will be described along with the rest of the CAOS commands.

Macro Commands

Creation Commands

Macro	Description
These are the commands that will create a new game presence within the world, when a NEW: object is created it will be TARG, until focus is changed to another object. Make sure that object creation is preceded with INST to make sure the process is uninterrupted. The following explain the values needed for various object creations: *imagefile* – 4 letter token representing the sprite filename *totalinsprite* – total number of images in the sprite file *numimages* – number of images belonging to this object *imagenumber* – offset of first image associated with this object *plane* – plot plane for this object between the range of 0=back, 9000=front	
NEW: SCEN *imagefile totalinsprite imagenumber plane*	Creates a scenery object – scenery objects do not need to have a unique classifier. **Command only**
NEW: SIMP *imagefile numimages imagenumber plane clone*	Creates a Simple Object. By default Simple Objects have no ATTRibutes or BeHaViouR defined, these must be set up after creation. *clone* – 0 is the value normally used and means that all identical objects can use the same image gallery. 1 creates a whole new image gallery for this object, normally only required if that particular object's image is going to be modified. **Command only**
NEW: CBTN *imagefile numimages imagenumber plane*	Create a call button object. **Command only**
NEW: COMP *imagefile numimages imagenumber clone*	Create a Compound Object. By default it has no parts, these must be set up after creation – Compound Objects need at least one part. **Command only**
NEW: PART *part# relx rely imageoffset plane*	Add a part to the current Compound Object. Should be used immediately after creating a Compound Object to add one or more parts to it. *part#* is the part number (0-9) *relx* and *rely* are the position of the part relative to part 0 *imageoffset* is the base sprite for this part relative to the first sprite **for this object** **Command only**
NEW: VHCL *imagefile numimages imagenumber*	Create a vehicle object. **Command only**
NEW: LIFT *imagefile numimages imagenumber*	Create a lift object – lift objects work in conjunction with call buttons to move between floors. **Command only**
NEW: BKBD *imagefile numimages imagenumber backgroundcolour chalkcolour aliascolour textx texty*	Create a blackboard object – blackboard objects are the only ones that can use the BBD: commands designed to teach words and concepts. *backgroundcolour*, *chalkcolour* and *aliascolour* are the colour numbers to use for plotting text. *textx* and *texty* are the co-ordinates to place the text plot area relative to part 0 **Command only**
TOKN *abcd*	Convert 4 characters into an integer. This is generally only used for the following commands to reference a string moniker, rather than the integer they require. **RValue only**

NEW: CREA *moniker sex*	Create a new-born creature. *moniker* is the moniker to use to locate the child's genome, this should already exist. *sex* is 1 if male, 2 if female and 0 if randomly determined. **Note:** the moniker must be supplied as an integer and not a string literal, this allows the moniker to be stored in an object variable. If an explicitly named genome is to be used then the TOKN command should be used. For example, NEW: CREA OV00 0 creates a randomly sexed creature from the moniker stored in ov00. NEW: CREA TOKN ADAM 0 creates a randomly sexed creature from the genome 'adam.gen'. **Command only**
NEW: GENE *mum dad child*	Create a new genome file from *mum*'s and *dad*'s (or just mum if dad=0) genomes, and store the new genomes moniker in the variable *child*. For example, NEW: GENE TOKN ADAM TOKN EVE_ OV00 will create a child from the genomes 'adam' and 'eve_' and store the moniker in the variable ov00. **Command only**
NEW: CBUB *imagefile numimages imagenumber stringID*	Create a new Compound bubble object, this is a specialisation of a compound object that has text boxes capable of displaying text from the in-game string table – specified by *stringID*. **Command only**
NEW: BBTX *textpart relx rely width*	**Create a new text part for the current compound bubble object.** *textpart* – 0…9 a unique identifier for the part *relx* – x position of text part relative to compound bubble *rely* – y position of text part relative to compound bubble *width* – width of the text. **Note:** The text will word-wrap to the width specified, so the height of the box will depend on the length of the string. **Command only**

Specific Pointers

Macro	Description
These commands return (and/or set) the specified object. i.e. TARG NORN will set the TARG object to be the currently selected creature. VA00-99 and OV00-99 can also be used to refer to objects.	
TARG TARG *object*	Sets/returns the current object of script focus. **Command and RValue**
OWNR	Object who owns the currently executing script or the currently selected creature if used in a non-event script (i.e. injected for immediate execution). **RValue only**
FROM	Object who caused event leading to execution of this script. **RValue only**
NORN	Sets/returns the currently selected creature. **RValue and LValue**
PNTR	The hand/pointer object. **RValue only**
_IT_	Object that owner creature was attending to. **Note**: Only OWNR's IT can be determined, not TARG's. **RValue only**
CARR	Object that's carrying OWNR. **RValue only**
EDIT	The contents of the edit buffer, set by using the EDIT command or shift clicking an object. **RValue and command.**
TCAR	Returns carrier of TARG. **RValue only**
OBJP	Legacy command from Creatures 1. Specific pointer to store ID of an object for later retrieval. Can now use normal variables instead. **RValue and LValue**

Object Property Commands

Macro	**Description**
These commands set/return an objects physical properties and can be used when creating an object, or during event execution.	
PUPT *pose x y*	Pick Up point. This is the location of the virtual 'hand' an object uses to hold another. Simple Objects only. pose = corresponding object pose. Specify –1 to define default pickup point. x y = relative co-ordinate. **LValue only**
PUHL *pose x y*	Pick Up handle. The virtual 'handle', i.e. the point by which an object is carried. Simple Objects only. pose = corresponding object pose. Specify –1 to define default pickup handle. x y = relative co-ordinate. **LValue only**
ACCG	Sets/returns the objects Acceleration due to Gravity. This equates to density and air resistance in real-world physics. Value is in pixels per tick per tick. **RValue and LValue**
AERO	Sets/returns the objects Aerodynamics. % of x and y velocity lost due to air resistance every tick. 0% infinite movement! **RValue and LValue**
REST	Sets/returns the objects Restitution. % of y velocity lost after collision. I.e. Bouncyness. 0% infinite bounce! **RValue and LValue**
SIZE	Sets/returns the minimum door value through which an object will pass. Door values are set up using a Room Editor. **RValue and LValue**
RNGE	Sets/returns the range an object can see. **RValue and LValue**
ATTR	Sets/returns an objects Attributes. See reference section for values. **RValue and LValue**
BHVR *click touch*	Sets/returns the reactions to clicks by mouse and activation requests by a creature. The values for *click* and *touch* are listed in the reference section **Command only**
WDTH	Returns objects width **RValue only**
HGHT	Returns objects height **RValue only**

General Object Commands

Macro	**Description**
VA00-VA99	Temporary script variables that only exist for the duration of the script execution. VA00 to VA09 map directly to VAR0 to VAR9. **RValue and LValue**
OV00 –OV99	Permanent script variables that persist across all scripts of an object. OV00 to OV02 map directly to OBV0 to OBV2. **RValue and LValue**
_P1_ , _P2_	Macro script parameters specified in command MESG WRT+. **RValue only**
CLS2 *family genus species*	Used when creating an object, this is how you assign it a unique classifier. Parameters specified as separate integer values., for example: SETV CLS2 2 10 300 would assign the current object to belong to family 2, genus 10 and species 300. **LValue only**
UNID	Unique Identifier. A UNID of zero is invalid and indicates that the object does not exist. **Note:** This command has been left in for completeness, because using TARG will produce the same results. **RValue only**
GRAV	Gravity. 1 = on, 0 = off. When an object is falling GRAV has a value of 1, set it to 1 to make it fall. **RValue and LValue**
WALL	Collision script specific parameter. WALL last collided with. 0 = left, 1 = right, 2 = up, 3 = down, 4 = unknown. **RValue only**
KILL *object*	Delete the object specified. i.e. KILL EDIT deletes the object in the EDIT buffer, KILL OWNR kills the owner of the script, KILL TARG kills the current TARG object. **Command only**
RELX	Relative X. Returns the relative position of object Targ from the script Owner object. **RValue only**
RELY	Relative Y. Returns the relative position of object Targ from the script Owner object. **RValue only**
TICK TICK *value*	Sets/returns the object timer script tick rate. Set TICK to 0 to stop the timer script executing. **Command and RValue.**
FRZN	Sets/returns whether an object is Frozen. Can be 1 or 0. When an object is frozen it no longer receives updates and, in the case of creatures, external stimuli. **RValue and LValue**
POSX	Position X. Returns the X co-ordinate of the centre point of object Targ. **RValue only**
POSY	Position Y. Returns the Y co-ordinate of the centre point of object Targ. **RValue only**
POSL	Returns the object's Position – Left edge of object **RValue only**
POSR	Returns the object's Position – Right edge of object **RValue only**
POSB	Returns the object's Position – Bottom edge of object **RValue only**
POST	Returns the object's Position – Top edge of object **RValue only**

LIML	Returns object's Limit to the Left as absolute value of rooms boundary **Note:** Room boundaries are not the same as walls, rooms can join but still allow full passage between them. **RValue only**
LIMR	Returns object's Limit to the as absolute value of rooms boundary **Note:** Room boundaries are not the same as walls, rooms can join but still allow full passage between them. **RValue only**
LIMB	Returns object's Limit to the Bottom as absolute value of rooms boundary **Note:** Room boundaries are not the same as walls, rooms can join but still allow full passage between them. **RValue only**
LIMT	Returns object's Limit to the Top – i.e. how far away from a room boundary **Note:** Room boundaries are not the same as walls, rooms can join but still allow full passage between them. **RValue only**
FMLY	Returns the object's family **RValue only**
GNUS	Returns the object's genus **RValue only**
SPCS	Returns the object's species **RValue only**
MOVS	Returns the Movement Status of an object. See reference section for values. **RValue and LValue**
ACTV	Returns objects Active status. See reference section for values. **RValue and LValue**
NEID	Returns an objects Neural ID – i.e. which attention lobe neurone corresponds to this object **RValue only**
TOTL *family genus species*	Returns the number of objects in the world who fit this specifier **RValue only**
TOUC *object1 object2*	Returns 1 if these two objects are in contact. e.g. DOIF TOUC TARG OWNR GT 0. **RValue only**
SLIM	Set the limits of the target object, this is needed when an object is first put into the world (or after a MCRT command) because it appears in a state where it's boundary limits are not defined. **Command only**
ADDV *value1 value2*	value1 = value1 + value2 **Command only**
SUBV *value1 value2*	value1 = value1 – value2 **Command only**
MULV *value1 value2*	value1 = value1 * value2 **Command only**
DIVV *value1 value2*	value1 = value1 / value2 **Command only**
MODV *value1 value2*	Performs a modulo arithmetic on the two values **Command only**
NEGV *value*	value = 0 – value **Command only**
ANDV *value1 value2*	value1 = 1 if value1 = 1 and value2 = 1 **Command only**
ORRV *value1 value2*	Performs a logical *OR* on the two values given **Command only**
RNDV *variable min max*	Set a *variable* to a random number between *min* and *max.* **Command only**

SETV *variable value*	Set a *variable* to have the value *value.* **Command only**
BBLE [*text*] *duration type location*	Create a speech bubble object containing the given *text* for the specified number of ticks. *duration* – the number of ticks to display the bubble for *type* – 0=speech bubble, 1=think bubble *location* – 0=track owner, 1=centre in viewport **Command only**

Execution Flow Commands

Macro	**Description**
These commands all alter the normal control flow through a macro script or allow you select a new TARG object.	
ENUM *family genus species* NEXT	Enumerate across all objects matching the specifier **Note:** All commands between ENUM and NEXT run as INST. TARG is returned to OWNR (in event scripts) after enumeration. **Command only**
ESEE *family genus species* NEXT	Enumerate across all specified objects that can be seen by the owner object. **Note** All commands between ESEE and NEXT run as INST. TARG is returned to OWNR (in event scripts) after enumeration. **Command only**
ETCH *family genus species* NEXT	Enumerate across all specified objects that are touching the owner object. **Note** All commands between ETCH and NEXT run as INST. TARG is returned to OWNR (in event scripts) after enumeration. **Command only**
ESCN *family genus species* NSCN	Enumerate all specified scenery objects. **Command only**
STAR *family genus species*	Can See Target Random. Pick a random object from those objects that match the given family genus species and can be seen by the Owner object. Implemented as a command only; the result is placed in object pointer Targ. **Note:** this command is computationally expensive. **Command only**
RTAR *family genus species*	Randomly selects a member that matches the specifier and sets it as TARG. Null if no member exists. **Command only**
INST	Make the rest of this macro execute in a single tick. This is used to ensure the creation of an object (for example) is not interrupted during the scheduling of concurrent scripts. **Command only**
SLOW	Negates previous INST command, returns control flow to normal execution. **Command only**
STOP	Stop execution of this macro script. **Command only**
ENDM	Compulsory command at end of macro script. **Command only**
SUBR *label*	Identifies a subroutine. *label* is a 4 character unique label that allows the subroutine to be called using GSUB **Command only**
GSUB *label*	Move execution to the given SUBR label. **Command only**
RETN	Returns from a GSUB **Command only**
REPS *n*	Repeat the following code *n* times, up to the next REPE. **Command only**
REPE	End of repeat loop. **Command only**
LOOP	Top of a LOOP-UNTL or LOOP-EVER statement. **Command only**

UNTL *value1 relation value2*	Part of a LOOP-UNTL statement. Repeat loop unless condition is true. Valid *relations* are: EQ – Equal NE – Not equal GT – Greater than LT – Less than GE – Greater than or equal to LE – Less than or equal to BT – Bit-wise *AND* function BF – Bit-wise *NAND* function **Command only**
EVER	Part of a LOOP-EVER statement. Repeat loop forever. **Command only**
DOIF *value1 relation value2*	Do next instruction of the condition is true, otherwise skip to after nested ELSE or ENDI. Valid *relations* are: EQ – Equal NE – Not equal GT – Greater than LT – Less than GE – Greater than or equal to LE – Less than or equal to BT – Bit-wise *AND* function BF – Bit-wise *NAND* function **Command only**
ELSE	If reach an ELSE because previous DOIF was true then jump to ENDI, otherwise DOIF was false so carry on processing from here. **Command only**
ENDI	Marks end of DOIF or DOIF-ELSE statement. **Command only**
WAIT *ticks*	Wait for the specified number of game *ticks* before continuing with next instruction. A tick is roughly 1/10 sec. **Command only**

Animation and Movement Commands

Macro	Description
These commands all alter the physical appearance or location of the TARG object.	
ANIM [*123456789R]* ANIM [*001002003R]*	Start animation of TARG object or part, using the poses specified. Animations ending in R will repeat indefinitely. **Note:** For objects the entries are single digit poses, for creatures the entries are **3** digit poses. Max length of animation string in both cases is 32 digits. **Command only**
OVER	Wait until TARG objects animation is over. **Note:** ANIMs ending in R will never end, hence causing a lock out. **Command only**
POSE POSE *number*	Sets/returns the objects (or parts) pose – i.e. which image number it is currently using. **RValue and command.**
PRLD [*1234*]	Pre-load image cache with these poses for smoother animation later **Note:** Not applicable on a creature **Command only**
BASE *number*	Specify the base image number for this object/part. Can be used to allow anims from a large table of images by moving base sprite number around table. This is the only way to use ANIM on an object with more that 10 images in its image gallery because ANIM uses a single digit to specify poses. **Command only**
VELX *velocity*	Sets/returns the object's velocity in X direction. **RValue and LValue**
VELY *velocity*	Sets/returns the object's velocity in Y direction. **RValue and LValue**
MVTO *x y*	Move object to absolute location and redraw **Note:** Should not be used on a creature that has had its limit set. **Command only**
MCRT *x y*	Move object to absolute location and redraw, this is different from MVTO in that the object has it's limits removed before moving. **Command only**
MVBY *x y*	Move object by relative amount and redraw. **Command only**

Object Communication

Macro	**Description**
These commands are used to send messages between objects and creatures. The possible values for *message* and *stimulus#* are detailed in the reference section. *list_of_stimulus_items* is comprised of the following items, *significance* – amount to nudge the Source lobe neurone by, this decides the significance of the object. *input* – which neurone in the General Sense lobe to fire (or 255 for none) *intensity* – the amount to nudge the General Sense neurone selected as *input* *features* - <<do not use non-zero values – not implemented >> *chem0 amount0* – this part lists 4 chemicals to emit and the amounts. If less than 4 chemicals are *chem1 amount1* - required then the remaining values should be set to 0 *chem2 amount2* *chem3 amount3*	
MESG SHOU *message*	Shout *message* to all creatures that can hear the object **Command only**
MESG SIGN *message*	Send *message* to all creatures that can see the object **Command only**
MESG TACT *message*	Send *message* to all creatures that are touching the object **Command only**
MESG WRIT *object message*	Write a *message* to a particular *object* (e.g. NORN, FROM, _IT_ etc) **Command only**
MESG WRT+ *object m p1 p2 delay*	Write message *m* with parameters *p1* and *p2* to *object*; to be handled in *delay* clock ticks. If *delay* = 0 the message will be handled immediately. *m* must be in the range 256..65535. (Messages 0..12 are currently used for ACTIVATE1 etc. events). The message that is sent will trigger event *m* on the *targ* object. **Command only**
STM# SHOU *stimulus#*	Shout *stimulus#* to all creatures within earshot **Command only**
STM# SIGN *stimulus#*	Send *stimulus#* to all creatures that can see the object **Command only**
STM# TACT *stimulus#*	Send *stimulus#* to all creatures that are touching the object **Command only**
STM# WRIT *object stimulus#*	Send *stimulus#* to the specified *object* **Command only**
STIM SHOU *list_of_stimulus_items*	Shout the *list_of_stimulus_items* to all within earshot **Command only**
STIM SIGN *list_of_stimulus_items*	Send the *list_of_stimulus_items* to all that can see the object **Command only**
STIM TACT *list_of_stimulus_items*	Send the *list_of_stimulus_items* to all that are touching the object **Command only**
STIM WRIT *object list_of_stimulus_items*	Send *list_of_stimulus_items* to a particular *object*. **Command only**

Room based Commands

Macro	**Description**
These commands are used to set/return details about the room system. This includes both the physical arrangements of rooms and also operations on the attributes of the room that TARG is in.	
TEMP	Sets/returns the temperature at targ. **RValue and LValue**

LITE	Sets/returns the Light Level at targ. **RValue and LValue**
RADN	Sets/returns the Radiation at targ. **RValue and LValue**
ONTR	Sets/returns the Organic nutrients at targ. **RValue and LValue**
INTR	Sets/returns the Inorganic nutrients at targ. **RValue and LValue**
PRES	Sets/returns the Pressure in room at object. **RValue and LValue**
WNDX	Sets/returns the wind X component at targ. **RValue only.**
WNDY	Sets/returns the wind Y component at targ. **RValue only.**
HSRC	Sets/returns the Heat source in room at targ. **RValue and LValue.**
PSRC	Sets/returns the Pressure source in room at targ. **RValue and LValue.**
LSRC	Sets/returns the Light source in room at targ. **RValue and LValue.**
RSRC	Sets/returns the Radiation source in room at targ. **RValue and LValue.**
RMNO	Room Number. Return the room number of the current room of Targ. **RValue only**
RTYP	Room Type. Sets/returns the type of the room at the centre point of Targ. Returns: -1 Invalid. 0 Under the Albian surface. 1 On the Albian surface. 2 In the sea. 3 In the sky. **RValue and LValue**
RMN# *room edge*	Room Neighbour count. Number of neighbours a room has on a given edge. *Room* = room number. *edge*: 0 = left, 1 = right, 2 = up, 3 = down. **RValue only**
RMND *room direction door*	Room Neighbour door Door-open value. *room* = room number. *direction*: 0 = left, 1 = right, 2 = up, 3 = down. *door* = number of door required. E.g. 1 for 1st, 2 for 2nd, etc. **RValue and LValue**
RMNR *room direction door*	Room Neighbour door Destination room ID. room = room number. direction: 0 = left, 1 = right, 2 = up, 3 = down. neighbour = number of door required. E.g. 1 for 1st, 2 for 2nd, etc. **RValue and LValue**
ROOM *room property*	ROOM property values. *room*: room ID. *property*: 0 = left, 1 = top, 2 = right, 3 = bottom, 4 = type, 5 = floor value, 6 = organic nutrient, 7 = inorganic nutrient, 8 = temperature, 9 = pressure, 10 = wind x component, 11 = wind y component, 12 = light level, 13 = radiation, 14 = heat source, 15 = pressure source, 16 = light source, 17 = radiation source, 18 = visited flag, 19 = drop status **RValue only**

ROOM *room p0 p1 … pn*	ROOM property values. *room*: room ID. *p0* = left, *p1* = top, *p2* = right, *p3* = down, *p4* = type, *p5* = floor value, *p6* = organic nutrient, *p7* = inorganic nutrient, *p8* = temperature, *p9* = pressure, *p10* = light level, *p11* = radiation, *p12* = heat source, *p13* = pressure source, *p14* = light source, *p15* = radiation source, *p16* – drop status.. **Command only**
DELR *room*	Delete Room. Remove the room from the map. *room*: room ID. **Command only**
DELN *room direction*	Delete Neighbour doors. Removes all neighbours for a room. Does not actually delete the rooms, just the relationship between room and the neighbours. *room*: room ID. *direction*: 0 = left, 1 = right, 2 = up, 3 = down. **Command only**
DOOR *direction RoomID1 RoomID2*	DOOR open value. Gets/sets the door open value between the given rooms. **RValue and LValue**
WLDW	World Width, in pixels. **RValue only**
WLDH	World Height, in pixels. **RValue only**
TECO	Toggle Ecosystem on and off. Switches the room system update mechanism on or off. Default: on. **Command only**
OBST *direction*	Obstacle.. Searches the room from the centre point of TARG for a floor or room door that presents an obstacle to the agent assuming it is moving in the given direction. The value returned is the distance to this obstacle. This macro command takes into consideration the SIZE of the agent, the floor value of the room and appropriate door values. If no obstacle is presented to the agent then a very large number is returned. direction: 0 = left, 1 = right, 2 = up, 3 = down. **RValue only**
OBDT *direction*	Obstacle Distance. Returns the distance to any obstacle in the given direction regardless of SIZE, floor value or door values. Direction: 0 = left, 1 = right, 2 = up, 3 = down. **RValue only**
OBSV *direction*	Obstacle Value. Returns the value of the door for the first obstacle in the given direction. Direction: 0 = left, 1 = right, 2 = up, 3 = down. **RValue only**
FLOR	Floor. Returns the y co-ordinate of the floor at the centre x position of Targ. **RValue only**
RMS#	Returns the number of rooms defined in the world. **RValue only**
GRND *x*	Returns the ground level of the first surface room at *x* position in the world. Ground, in this case, is just the bottom of the first surface room and doesn't take into account uneven floors or door values of adjoining rooms. **RValue only**
ISAR *room#*	Returns whether the given room number exists, returns 1 if true 0 if false. *room#* - the number of the room to check **RValue only**

World Commands

Macro	Description
These commands return information about the state of certain world elements.	
SEAN	Returns the season Number: 0..3. (Spring, Summer, Autumn, Winter) **RValue only**
SEAV	Season Value: 0..19. Returns the position within the season. **RValue only**
ASEA	Advance Season. Move to the beginning of next season. **Command only**
TMOD	Time Of Day: 0..4. Dawn, Morning, Afternoon, Evening, and Night. **RValue only**
YEAR	YEAR. Returns the current year in Albia. Starting with 0000 AD (first chronicled year After Disaster) **RValue only**
EGGL	Egg limit. Returns the number of norns above which eggs should stop hatching **RValue only**
HATL	Hatchery limit. Returns the number of norns above which the hatchery should shut down. **RValue only**

Compound Object Commands

Macro	**Description**
These commands are only to be used on Compound Objects and allow you to define different click areas on the object that will trigger events. Compound Objects are different from Simple Objects in that they can not be picked up – even if ATTR is set to allow it. Compound objects are generally large objects and have the ability to be made up of parts, with each part able to animate independently of any other part.	
SPOT *spot# left top right bottom*	Set up a compound object hotspot for users/creatures to click on. These commands (along with KNOB) allows different parts of an object to produce different events when clicked on. *spot#* = (0-5) a number to identify this particular spot *left/top/right/bottom* = Co-ordinates of hotspot relative to part 0 of the object. Set *left/top/right/bottom* to –1 –1 –1 –1 to remove/blank a hotspot **Command only**
KNOB *function hotspot#*	Attach compound objects' hotspot to a particular function. *function* specifies the event and usage 0=creature activate 1 1=creature activate 2 2=creature deactivate 3=pointer activate 1 4=pointer activate 2 5=pointer deactivate *hotspot#* refers to the hot spot number defined with the SPOT command. **Command only**
KMSG *knob flag message*	Knob Message. Set the object message associated with the knob. This allows messages other that act1,act2 and deac to be sent – and also allows the usage to be altered (e.g. creature only) *knob* – (0-5) and refers to the KNOB *functions* already set up *flag* – used to describe the usage of the knob 1=Only creature can use this knob 2=Only pointer can use this knob 3=Both creature and pointer can use this knob *message* – message/event to be sent when this knob is activated. Values are (0-65535) **Command only**
PART *number*	Set the part *number* for future actions to work on (Compound Objects). e.g. animations **Command only**

Vehicle/Lift Object Commands

Macro	**Description**
These commands are particular to objects of type LIFT or VHCL – these are the only objects which can carry a creature around the world	
CABN *left top right bottom*	Set the relative co-ordinates for a vehicle cabin, the place where creatures are held. **Command only**
DPS2 *gravity*	Drop Passengers 2. Causes any creatures carried by the vehicle to be 'dropped'. *gravity* 0 = gravity is not activated on the object and must be done manually under script control. non-zero = gravity is activated on the object **Command only**
DPAS	Drop passengers. This works the same way as specifying DPS2 1. i.e. all passengers are ejected and have gravity operational. **Command only**
GPAS	Get passengers. Loads any nearby creature into the vehicles cabin **Command only**
SPAS *vehicle creature*	Loads a named *creature* into the named *vehicle* **Command only**
LACB	Align Lift cabin with room floor. If this *value* is set to non-zero the lift will stop at a position where the bottom of its cabin is aligned with the room floor. The default behaviour is to align the bottom of the lift with the floor. **LValue and RValue**
XVEC *value*	Sets/returns the vehicles x movement vector in 1/256th of a pixel. This command is a legacy from Creatures 1 – VELX will work just as good. **RValue and LValue**
YVEC *value*	Sets/returns the vehicles y movement vector in 1/256th of a pixel. This command is a legacy from Creatures 1 – VELX will work just as good. **RValue and LValue**
BUMP	Returns a vehicle's collision data. This is a legacy command left in for completeness. To utilise this a vehicle must move use XVEC and YVEC (rather than VELX and VELY) and have ATTR set up to respond to boundaries. The values returned are from a bit-field where: bit0 – hit left bit1 – hit right bit2 – hit top bit3 – hit bottom So, for example: a value of 1 corresponds to having hit a wall to the left; a value of 4 corresponds to having hit a ceiling; a value of 5 corresponds to having hit a ceiling and a wall to the left. **RValue only.**
TELE *x y*	Teleport the vehicles occupants to the location *x y* and move the camera too. **Command only**

Blackboard Commands

Macro	**Description**
These commands apply to TARG, which must be a blackboard object. Blackboard objects have a word list associated with them – this list needs to be set up for each computer object. The current word of a blackboard is always pointed to by the value of ov00. i.e. to point to the word at index 5, set ov00 to 5. The default word list is in the reference section.	
BBD: VOCB *bs vs cnt*	Set Blackboard Vocabulary. This is used to move words from the game word list into a specific blackboard objects word list. bs = Start position in blackboard word list, vs = Start position in game vocabulary list, cnt = Number of words to copy. **Command only**
BBD: VCB1 *wi vi*	Set BlackBoarD VoCaBulary 1 word. This is used to move a single word from the game word list into a specific blackboard objects word list. wi = index in blackboard word list, vi = index in game vocabulary list. **Command only**
BBD: WORD *index ID [text]*	Install a single named word into the blackboards word list. *index* is the index in the blackboard word list *ID* is the word concept ID number *[text]* is the text to be used for the word **Command only**
BBD: SHOW *n*	*n*=0:Wipes text from blackboard *n*=1:Draws the current text string text (ov00) onto part 0 **Command only**
BBD: EMIT *n*	Broadcasts the current word so that nearby creatures can learn the association between text and concept *n*=0:The word will be broadcast as if it were read – to creatures that are looking at the blackboard *n*>0:The word will be broadcast as an audible lesson – to creatures that are in earshot of the blackboard **Command only**

Compound Bubble Object Commands

Macro	Description
All of these commands are for the Compound Bubble objects created with NEW: CBUB	
BBTX *textpart string#*	Associate a text string in the application resource table with the given text part. *textpart* – a unique identifier for the part *string#* - index into the string table **Command only**
BBT2 *textpart* [*string*]	Specify a text string to be used for the given *textpart* **Command only**
BBFD *textpart r g b*	Specifies the target colour of the text in the given *textpart*. The bubble will then fade the text into this colour. *r* – red component 0…255 *g* – green component 0…255 *b* – blue component 0…255 **Command only**

Call Button Commands

Macro	Description
Call Buttons should be created immediately after the lift for which they operate – in a top to bottom order. The following commands allow the call buttons to set up the floor levels at which the lifts stop.	
CBRG	Call Button register. Notifies the last created lift of this call button's position. This sets the floor level of the lift to be the room surface at this position. **Command only**
CBRX *x y*	Call Button register extended. Notifies the last created lift of this call button and the specified position. This sets the floor level of the lift to be the room surface at the specified position. **Command only**

Rain Cloud Object Commands

Macro	Description
A rain cloud object is something that can generate a tiling image underneath it – these were not used in the release version of Creatures 2 but have been left in for completeness. The use of this type of object relies on the fact that there is a sprite file called RAIN.S16 which contains 12 images of size 64x64. If this file does not exist then strange things can happen. The NEW: command is used to create a rain cloud object and the RAIN command is used to start/stop the effect and change it's appearance.	
NEW: RAIN *imagefile numimages imagenumber plane clone*	Creates a new rain cloud object, these are like Simple Objects but have a weather effect underneath. *imagefile* – the sprite file to use for the cloud itself *numimages* – the total number of images to be used for this object *imagenumber* – the number of the first image belonging to this object *plane* – the image plane to plot the object at. *clone* – create a cloned image gallery yes/no **Command only**
RAIN *property* RAIN *property value*	Get/set a *property* of the weather effect of a RAIN object. 0=x position, relative to object 1=y position, relative to object 2=width 3=height 4=angle of effect (updated by wind) 5=speed – number of pixels moved per update 6=mode – (0=stopped, 1=running, 2=start!, 3=stop!) 7=rain image number (0-11) **Command and RValue.**

Camera Commands

Macro	Description
These commands all move the game window (camera) or return information about it.	
SYS: CMRP *x y*	CaMeRa Pan. Similar to the macro SYS: CMRA. Moves the camera to the specified x and y position. If that location is within the window view then the camera smooth scrolls to this position, otherwise the camera jumps to that location. This command does not effect the users choice of scrolling mode. **Command only**
SYS: CMRA *x y*	Position camera with this co-ordinate as the top-left **Command only**
SYS: CAMT	Positions camera to point at current TARG **Command only**
CMRX	CaMeRa centre X co-ordinate. The centre x co-ordinate of the camera in world co-ordinates. **RValue only**
CMRY	CaMeRa centre Y co-ordinate. The centre y co-ordinate of the camera in world co-ordinates. **RValue only**
SYS: WPOS *x y width height*	Attempt to position the game window, with the top left corner being at *(x,y)* of the screen and the window having dimensions *width* and *height*. **Command only**

Music and Sound Effect Commands

Macro	**Description**
These commands alter the in-game ambient music and also play spot effects. Where a command requires a *filename* it has to be a 4-digit name with the extension omitted. For example, to play a sound effect called 'TEST.WAV' from the C2 directory you could use: SNDE TEST	
THRT	Sets/returns the THReaT, how dangerous an object is to a creature. This attribute is used only for the in-game music system. 255 = high threat; 0 = no threat. **RValue and LValue**
RMSC *x y [track name]*	Room MuSiC. Specify music track associated with the room at this *x y* co-ordinate. [] (Empty string) = no change in track [Silence] = play no track. **Command only**
RCLR	Room CleaR music. Resets the music track of all rooms to "no change". **Command only**
MUSC *property*	MUSiC manager property. Property: 0 = Volume (0 – 255) 1 = Mood (0 – 255) 2 = Threat (0 – 255) 3 = Automatically update music according to creature mood and threat. (0, 1). **RValue and LValue**
SNDE *filename*	Play sound effect if TARG is visible on screen. **Command only**
SNDQ *filename delay*	Play sound effect after a short delay if TARG is visible on screen. **Command only**
SNDC *filename*	Play controlled sound effect if TARG is visible on screen. A controlled sound will follow TARG around as it moves, for the duration of the effect. **Command only**
SNDL *filename*	Start a controlled sound loop if TARG is visible on screen **Command only**
PLDS *filename*	Preload the named sound into the sound cache if TARG is visible or just off screen. **Command only**
STPC	Stop any controlled sounds currently playing with this TARG. **Command only**
FADE	Fade out any controlled sound currently playing with this TARG. **Command only**

Creature Commands

Macro	**Description**
All these commands apply to the TARG object which must be a creature	
DRIV *n*	Returns the state of the creatures Drive number *n* **RValue only**
DRV!	Returns the number of the creatures most pressing drive **RValue only**
CHEM *chemical* CHEM *chemical amount*	Returns the concentration of a chemical in a creatures bloodstream or adds the amount given. **Command and RValue**
BABY	Moniker of child genome if creature is pregnant. Set to 0 to finish pregnancy or a moniker to make pregnant. **RValue and LValue**
ASLP	Sets/returns whether a creature is asleep. When asleep the eyes close and the creature becomes insensible to some stimuli. A change of action automatically wakes creature up again. 0=awake, 1=asleep **RValue and Command**
DEAD	Returns whether a creature is dead or alive 0=alive, 255=dead **RValue only**
UNCS	Returns whether a creature is unconscious or not 0=conscious, 1=unconscious **RValue only**
INS#	Returns the number of instincts the creature still has left to process. Instincts are processed in the egg and during sleep. **RValue only**
DIRN	Returns the direction the creature is facing. 0=North (back to viewer) 1=South (facing viewer) 2=East 3=West **RValue only**
MONK	Returns the moniker of the creature **RValue only**
CREA *moniker*	Returns the creature with the specified moniker **RValue only**
ORGN	Returns the number of organs in the creature **RValue only**
INJR *organ amount*	Reduce the life force of the specified organ. *Amount* is in the range of (0-255) and max damage is a 1/10 of the organs initial life force. *Organ* is –1 for a randomly selected organ and 0 for the body organ. **Command only**
FIRE *x y amount*	Fire the neurone whose position is at *xy* – *amount* is the signal strength (0-255) **Command only**
TRIG *lobe cell amount*	Fire the neurone that is at the *lobe* and *cell* numbers specified – *amount* is the signal strength (0-255) **Command only**
APPR	Approach _IT_. Chose a walking gait according to chemo-receptors, then start walking towards _IT_. Continue with next instruction when you are within reach **Command only**
WALK	Walk indefinitely. Chose a walking gait according to chemo-receptors, then start walking. If extraspective you'll continuously walk towards _IT, but the command is primarily for introspective walking. **Command only**

TOUC	Reach out and touch _IT_, normally preceded with an APPR macro. Continue with next instruction when you have successfully touched _IT_. If total failure (object gone away) then present action schema is suppressed. **Command only**
POIN	Point to _IT_. As for TOUC but creature reaches out to object with head facing the camera. **Command only**
AIM: *act*	Set the target point on the _IT_ object for subsequent APPR and TOUC commands. *act* is the action number to decide which spot to aim for 0=activate 1 1=activate 2 2=deactivate **Command only**
SAY# *number*	Speak word *number* in a speech bubble, and send that word as a signal message to creatures in earshot **Command only**
SAY$ [*string*]	Speak the given *string* in a speech bubble – no signal sent **Command only**
SAYN	Speak your most pressing need **Command only**
IMPT *value*	Signify how important this (voluntary) action is. i.e. how unlikely it is that another action will override this one before it has finished. *value* is the amount that gets used to nudge the current decision neurone and should be a low number. This instruction should be used at the start of every creature action script, and may used within a script if the importance changes during a later phase. **Command only**
DONE	This voluntary or involuntary action has been completed. For voluntary actions the decision neurone is reset to force the creature to make a new decision. Put this command at the end of any transient voluntary action (e.g. activate 1 but not walk east) and after every involuntary action. **Command only**
LTCY *action min max*	Set the latency for the creatures given involuntary action (only relevant to involuntary actions). Prevent this action from repeating for an amount of time defined by *min max*. Delay is a random amount of ticks between *min**4 and *max**4. This command may be called at the end of an involuntary action script to prevent reactivation until the chemical that triggered the action has subsided. **Command only**
DREA *max*	Start dreaming. This starts processing of any pending instincts instead of receiving sensory data from the environment. Normally this should only be done during deep sleep and during embryology. *max* pending instincts will be processed and then dream state switches off automatically. **Command only**
DROP	Drop any objects you are carrying **Command only**
MATE	Only relevant to male creatures – pass on waiting sperm to _IT_ (if _IT_ is female and the same genus). Female will conceive if she is in the right condition – fertile and receptive. **Command only**
SNEZ	Creature sneezes – this will infect any nearby creatures or environment with any live bacteria it has in it. **Command only**
CAMN	Returns the creatures age in minutes. **RValue only**

CAGE	Returns the creatures stage of life. 0 – embryo 1 – baby 2 – child 3 – adolescent 4 – youth 5 – adult 6 – old 7 – senile **RValue only**

DDE commands

Macro	**Description**
These commands return a value into a DDE command window. An application that communicates with Creatures 2 via OLE is needed to view these results.	
DDE: GIDS ROOT	Returns a list of all the Family numbers that exist, separated with a space. **Command only**
DDE: GIDS FMLY *number*	Returns a list of all genuses that exist from within a certain family *number*. **Command only**
DDE: GIDS GNUS *family number*	Returns a list of all species that exist from within a certain *family* and genus *number* **Command only**
DDE: GIDS SPCS *family genus number*	Returns a list of all events that exist from within a certain *family genus* and species *number* **Command only**
DDE: GETB ALLR *property*	ALL Rooms property value. Returns a particular property for all rooms. 0 = left, 1 = top, 2 = right, 3 = bottom, 4 = type, 5 = floor value, 6 = organic nutrient, 7 = inorganic nutrient, 8 = temperature, 9 = pressure, 10 = wind x component, 11 = wind y component, 12 = light level, 13 = radiation, 14 = heat source, 15 = pressure source, 16 = light source, 17 = radiation source, 18 = visited flag. **Command only**
DDE: GETB RPTY *room#*	Returns properties of specified *room#* as a string. Results are in the following order: Temperature Pressure Wind X Wind Y Light level Radiation Organic Nutrients Inorganic Nutrients **Command only**
DDE: GETB RRCT	Room ReCTangles returns the ID, RECT structure and VISITED flag for every room. Returns: ID | top | left | bottom | right | VISITED (0 = NO, 1 = YES) | Ampersand as room data separator | **Command only**
DDE: GETB NEWV	NEWly Visited rooms. Returns a list of room IDs for all rooms visited since the last query. **Command only**
DDE: GETB LVOB *f g s*	Locate Visited Objects. Returns a list of X / Y positions for all the objects matching <f, g, s> that are in rooms that have been visited. After each objects details is an ampersand to allow later expansion if needed. f-family g-genus s-species Returns: Xvalue | Yvalue | and | **Command only**
DDE: GETB BIOC	Get biochemistry buffer. Returns a string of the target creatures chemical concentrations in the format "123|33|0|…|". The string contains 256 values – as there is one entry for every chemical slot. **Command only**

DDE: GETB ORGN	Get organ buffer. Returns an integer specifying how many organs the target creatures biochemistry contains followed by the data for each organ in the format "1|2|4|55|". The sequence of organ data is as follows: Clock Rate (byte) Clock (byte) BioTick (unsigned integer) LifeForce (byte fraction of short term lifeforce) Initial LifeForce (float) Short Term LifeForce (float) Long Term LifeForce (float) Long Term Rate Of Repair (integer) Long Term Rate Of Repair Modifier (integer) Injury To Apply Next Activation (integer) EnergyCost (integer) Damage Due To ZeroEnergy (integer) Receptor Count (integer) EmitterCount (integer) ReactionCount (integer). **Command only**
DDE: GETB EMTR *organ-number*	Get Emitter Buffer. Returns an integer specifying how many emitters the Targ creature's specified organ contains followed by the data for each emitter in the format "1|2|33|". The sequence of organ data is as follows: Locus organ identifier (byte) Locus tissue identifier (byte) Locus specifier (byte) Chemical emitted (byte) Emission threshold (byte) Emission rate (byte) Gain (byte) Effect flags (byte) Current value of source locus (byte). **Command only**
DDE: GETB *organ-number*	Get Receptor Buffer. Returns an integer specifying how many receptors the Targ creature's specified organ contains followed by the data for each receptor in the format "1|2|33|". The sequence of organ data is as follows: Locus organ identifier (byte) Locus tissue identifier (byte) Locus specifier (byte) Chemical monitored (byte) Reception threshold (byte) Gain (byte) Effect flags (byte) Current value of destination locus (byte). **Command only**
DDE: GETB RCTN *organ-number*	Get Reaction Buffer. Returns an integer specifying how many reactions the Targ creature's specified organ contains followed by the data for each reaction in the format "1|2|33|". The sequence of organ data is as follows: Proportion of reactant 1 (byte) Chemical - reactant 1 (byte) Proportion of reactant 2 (byte) Chemical – reactant 2 (byte) Rate of decay (byte) Proportion of product 1 (byte) Chemical - product 1 (byte) Proportion of product 2 (byte) Chemical – product 2 (byte) **Command only**

DDE: PIC2 *width height [filename]*	PICture. Replaces macro PICT. This takes a snapshot of the currently selected creature of size *width* and *height* and saves it in the filename specified. The format of the file is S16. Note that the picture will be clipped by the top and bottom edges of the world and by the viewport. This will result in an image smaller than requested. Returns a string "1" if successful, "0" if not. **Command only**
DDE: SCRP *family genus species event*	Fetch a script from the scriptorium matching this specifier. **Command only**
DDE: PUTV *RValue*	Display the value of *RValue* **Command only**
DDE: NACT	Neural activity. Returns pipe-delimited numbers. First number is the number of brain lobes, each lobe then has an entry of the format "1|2|3" , where the sequence is in the following order: Number Of Cells Cell Flags Fire Count Total Output Total State **Command only**
DDE: LNEU *flags*	Returns info about all firing cells in the brain. All numbers returned by this command are scaled to range from 0 to 63 then added to ascii '0' to keep them in printable range. Each firing neurone is returned in the following format: XYS[den0info][den1info] X and Y is the neurone position, and S is the neurone state. Den0info and Den1info are only sent depending on the flags field: *flags:* 0 = output type 0 dens 1 = output type 1 dens The dendrite info format begins with a number indicating how many dendrites are feeding the cell, followed by a list of dendrites in the format: XYS Where X,Y and S are the position and state of the source neuron. **Command only**
DDE: LCUS *type organ tissue locusID*	Allows interrogation of creature loci values. *type* - 0=receptor loci 1=emmiter loci Details of the other values are listed in the reference section. **Note:** *organ* field does not refer to the organs defined in the genome. **Command only**
DDE: PUTS [*literal string*]	Send a string to the output window. **Command only**

DDE: GETB DATA	Gets all creature data in the following format. Each entry is separated with '|' Moniker Name Mothers Moniker Mothers Name Fathers Moniker Fathers Name Birth date Birth place Owner Name Owner Telephone Number Owner Address Owner Email State of Creature – 0=ok, 1=dead, 2=exported Gender Age **Command only**
DDE: GETB CNAM	Get creature's name. **Command only**
DDE: GETB CTIM	Get the amount of time the creature has been alive. **Command only**
DDE: GETB MONK	Get the creatures moniker in the form '464f4536' rather than '6EOF' **Command only**
DDE: GETB OVVD	Get observation data. This returns the following information, with each field separated with a '|' and each creature seperated with a '&': Name Moniker Sex – 1=male, 2=female Age – in "hours:mins" Pregnancy – either "N/A", "No" or number Life Force – either number terminated in % or "Dead" Medical – either "Healthy", "Sick" or "Dead" Room – number of room they are in Xpos Ypos **Command only**
DDE: PUTB [*literal string*] DATA	Set all the creatures details, see GETB DATA for ordering of details. **Command only**
DDE: PUTB [*literal string*} CNAM	Set the creatures name from the string. **Command only**
DDE: NEGG	Update number of natural eggs in the world. **Command only**
DDE: PANC	Pan camera to the creature. **Command only**

DDE: LOBE	Output locations and information about brain lobes. The first number returned is the number of lobes, and then the information about them is returned in the following format, with the '|' symbol separating fields: X position start Y position start Width Height Flags Number of Dendrites Dendrite type 0 source Dendrite type 0 minimum Dendrite type 0 maximum Dendrite type 1 source Dendrite type 1 minimum Dendrite type 1 maximum **Command only**
DDE: GENE	Output the number of genes for the following 14 types: Brain Lobe Chemical receptor Chemical emitter Chemical reaction Chemical half life Chemical starting concentration Organ Creature stimulus Creature genus Creature appearance Creature pose Creature gait Creature instinct Creature pigment **Command only**
DDE: WORD *index*	Read a word from the TARG blackboards word list. Sends the response of '###|text|' where '###' is the vocabulary slot for the idea represented by the blackboard picture whose index is *index*, and 'text' is the word it knows for it. **Command only**
DDE: CELL *lobe cell dentype*	Get statistics about this neurone, in the following format: Output State Number of dendrites of that type Total susceptibility Total short term weight Total long term weight Total strength The dendrite values are totalled from all dendrites of the given *dentype*, for a numbered *lobe* and *cell*. **Command only**
DDE: DIED	Update the number of norns that have died in the world **Command only**
DDE: LIVE	Update the number of natural born norns in the world. **Command only**
DDE: HATC	Update the number of hatchery norns in the world. **Command only**

Debug Commands

Macro	Description
These debug commands will display in the LOG window of Creatures 2 – only available in developer releases	
DBUG *rvalue*	All commands after this will run in an INST, sends the value of *rvalue*. **Command only**
DBGV *rvalue*	Sends the value of *rvalue*. Same as DBUG but doesn't force all commands to run in an INST. **Command only**
DBGM [*string*]	Displays *string* as a trace message **Command only**

System Commands

Macro	Description
These commands work at a level above the in-game operations and allow for such operations as pausing the game, altering the scriptorium etc.	
SCRP *family genus species event*	Indicates that the rest of this macro is to be installed into the scriptorium, making it available as a new/replacement event script for a given type of object. *family genus species* – indicate the owner of this script, if values of 0 are used for any of these then the script will be installed as a default script for a wide range of objects. i.e. if species is 0 then the script will apply to all who share the same family and genus. *event* – indicates the event that will invoke this script, see the reference section for a list of event numbers. **Command only**
SCRX *family genus species event*	Removes the script matching this description from the scriptorium. **Command only**
PAUS	PAUSe game. 0=normal, 1=pause Note that Creatures already calls a dispatch method for each applet currently running when the game pauses. **RValue and LValue**
GAME *category variable [state]*	GAME flow state. Sets or queries game flow state. Category: 0 = Applet custom condition, Variable: Applet slot number. Category: 1 = Applet advanced option, Variable: Applet slot number. 5 for Science Kit, 6 for Neuroscience Kit. Category: 2 = Game feature enable. Variable: 0 = Select Ettins/ Grendels, 1 = Infinite scroll **RValue and LValue**
LOCK	Prevents the execution of other event scripts taking place on this object and hence interrupting this script. Note that this will result in *lost* events and messages. Any script that contains a LOCK should also contain an UNLK. This will not prevent the execution of a SCRIPTEXCEPTION macro script however. **Command only**.
UNLK	Enables the execution of other event scripts on this object. This should be paired with command LOCK. **Command only**

EVNT *targ*	Adds an event label to the Event Bar, the three types possible are BIRTH, DEATH and EGG. To get a death event call this macro with *targ* as OWNR in it's death script. To get a birth event call this macro with *targ* as a new creature To get an egg event call this macro with *targ* as a newly created egg object. **Command only**
RMEV *targ*	Removes an event label from the event bar. The value for *targ* works the same as EVNT. **Command only**
SCOR *item*	Returns chronicle information concerning a particular game *item.* The values for *item* are: 0 – Generation 1 eggs used 1 – Number of natural eggs laid 2 – Number of deaths 3 – Number of creatures imported 4 – Number of creatures exported 5 – Number of creatures alive **RValue only**
HOUR	Returns the number of game hours elapsed since game start. **RValue only**
MINS	Returns the number of game minutes elapsed since game start. **RValue only**
SYS: EDIT *left top right bottom*	Draw a rectangle on the game window with the corners given by the co-ordinates *left top right bottom.* **Command only**
SYS: DMAP	Draw map. 1 -> draw room box, floor and vehicle cabins over backdrop. 0 -> do not draw room data. **Command only**
SYS: WTOP	Set the world window to be the foreground window. **Command only**
SYS: QUIT	Saves the world and closes the game window **Command only**
SYS: ABRT	Abandons changes to world and closes the game window **Command only**
SYS: WRLD *world*	Saves the current world and opens the named one **Command only**
SYS: CONV *filespec p*	CONVert sprite file. Command converts a given sprite file (located in the images directory) to the current system image format. If the file is already in the correct format it is not modified. *Filespec* – a 4 letter sprite file specifier. *p* – *p*=0 no progress dialog is displayed. *p*=1 a progress dialog is displayed. **Command only**
RNDR	ReNDereR image format. Returns 0 if image format is 555 and 1 if it is 565. **RValue only**
VRSN VRSN *number*	Return the Creatures Build ID or Only run this script if the Creatures Build ID is equivalent or higher than *number*. **LValue and Command**

LANG	Primary LANGuage of Windows desktop. Values are (in hexidecimal) as follows: Neutral 0x00 Chinese 0x04 Czech 0x05 Danish 0x06 Dutch 0x13 English 0x09 Finnish 0x0b French 0x0c German 0x07 Greek 0x08 Hungarian 0x0e Icelandic 0x0F Italian 0x10 Japanese 0x11 Korean 0x12 Norwegian 0x14 Polish 0x15 Portuguese 0x16 Russian 0x19 Serbo Croatian 0x1a Slovak 0x1b Spanish 0x0a Swedish 0x1d Turkish 0x1F **RValue only**

LNG+	Sub LaNGuage of Windows desktop. Values are (in hexidecimal) as follows: Neutral 0x00 Default 0x01 System Default 0x02 Chinese (Simplified) 0x02 Chinese (Traditional) 0x01 Dutch 0x01 Dutch (Belgian) 0x02 English (U.S.) 0x01 English (U.K.) 0x02 English (Australian) 0x03 English (Canadian) 0x04 English (Irish) 0x06 English (New Zealand) 0x05 French 0x01 French (Belgian) 0x02 French (Canadian) 0x03 French (Swiss) 0x04 German 0x01 German (Swiss) 0x02 German (Austrian) 0x03 Greek 0x01 Icelandic 0x01 Italian 0x01 Italian (Swiss) 0x02 Japanese 0x01 Korean 0x01 Norwegian (Bokmal) 0x01 Norwegian (Nynorsk) 0x02 Portuguese 0x02 Portuguese (Brazilian) 0x01 Serbo Croatian (Latin) 0x01 Spanish (Castilian)1 0x01 Spanish (Mexican) 0x02 Spanish (Modern)1 0x03 Turkish 0x01 **RValue only**

COB specific commands

Macro	Description
These commands are not true CAOS commands in that the in-game interpreter will not understand them; they are for use within a COB compiler to mark certain sections of script as performing a certain function.	
ISCR	Marks the following chunk of script as being for the installation of an object. The presence of SCRP, RSCR or ENDM is used to delimit the installation chunk
RSCR	Marks the following chunk of scripts as being for the removal of an object. The presence of SCRP, ISCR or ENDM is used to delimit the removal chunk

Reference

Event numbers

Event Number	Event Name	Description
These are the in-built events that agents can respond to – in the case of creatures their brain makes a decision to cause one of these events on some object (including themselves) or their actions cause one of these events to occur. Objects are scripted to respond to event messages and their behaviour comes from this event mechanism, for which reason there is a mechanism to make your own user defined object events.		
0	Deactivate	An action generally used to disable an active object – STOP to a creature
1	Activate 1	An action performed on an object – PUSH to a creature
2	Activate 2	An action performed on an object – PULL to a creature
3	Hit	An action performed on an object – HIT to a creature
4	Pickup	The event that is run when an object is picked up – GET to a creature
5	Drop	The event that is run when an object is dropped – DROP to a creature
6	Collision	The event that is run when an object has collided with a room boundary or floor
7	Enter scope	The event an object receives when the world starts up
9	Timer	The event an object receives when it's *tick* has reached zero
10	Push left	An action performed on an object
11	Push right	An action performed on an object
12	Eat	An action performed on an object
16	Extra Quiescent	CREATURE - Stand and watch it
17	Extra Activate 1	CREATURE – Activate 1 it
18	Extra Activate 2	CREATURE – Activate 2 it
19	Extra Deactivate	CREATURE – Deactivate it
20	Extra Seek	CREATURE – Go up and look at it
21	Extra Avoid	CREATURE – Walk/run away from it
22	Extra Pickup	CREATURE – Pick it up
23	Extra Drop	CREATURE – Drop anything you are carrying
24	Extra Need	CREATURE – Say what's bothering you
25	Extra Rest	*No such action needed – do not script*
26	Extra West	CREATURE – Walk idly to west
27	Extra East	CREATURE – Walk idly to east
28	Extra Eat	CREATURE – Eat it
29	Extra Hit	CREATURE – Hit it
32	Intro Quiescent	CREATURE – Stand and twiddle your thumbs
33	Intro Activate 1	*No such action needed – do not script*
34	Intro Activate 2	*No such action needed – do not script*
35	Intro Deactivate	*No such action needed – do not script*
36	Intro Seek	*No such action needed – do not script*
37	Intro Avoid	*No such action needed – do not script*
38	Intro Pickup	*No such action needed – do not script*
39	Intro Drop	CREATURE – Drop anything you are carrying
40	Intro Need	CREATURE – Say what's bothering you
41	Intro Rest	CREATURE – rest or sleep
42	Intro West	CREATURE – Walk idly to west
43	Intro East	CREATURE – Walk idly to east
44	Intro Eat	*No such action needed – do not script*
45	Intro Hit	*No such action needed – do not script*
50	Pointer Activate 1	POINTER – Left button click causing an Activate 1
51	Pointer Activate 2	POINTER – Left button click causing an Activate 2
52	Pointer Deactivate	POINTER – Left button click causing a Deactivate

53	Pointer Pickup	POINTER – Right button click, grab object
54	Pointer Drop	POINTER – Right button click to drop a held object
55	Pointer Push Left	POINTER – Left button click in push pointer mode
56	Pointer Push Right	POINTER – Left button click in push pointer mode
64	Involuntary 0	*CREATURE - Chemically invoked reflex or pathological behaviour*
65	Involuntary 1	*CREATURE - Chemically invoked reflex or pathological behaviour*
66	Involuntary 2	*CREATURE - Chemically invoked reflex or pathological behaviour*
67	Involuntary 3	*CREATURE - Chemically invoked reflex or pathological behaviour*
68	Involuntary 4	*CREATURE - Chemically invoked reflex or pathological behaviour*
69	Involuntary 5	*CREATURE - Chemically invoked reflex or pathological behaviour*
70	Involuntary 6	*CREATURE - Chemically invoked reflex or pathological behaviour*
71	Involuntary 7	*CREATURE - Chemically invoked reflex or pathological behaviour*
72	Death	CREATURE – Death event
200	Donate Sperm	MALE CREATURE – event that can triggered by a female during mating.
255	Exception	A script was referencing an object that has been destroyed
256	*user defined*	*All events from 256 – 65535 are user defined. This allows objects to employ common methods across different objects. The events can be triggered using the MESG WRT+ command and/or the KMSG command.*
…	*user defined*	
65535	*user defined*	

Values for ATTR

Value	Meaning	Description
1	Carryable	Creature or another object can pick up object
2	Mousable	Mouse can pick up object
4	Activateable	Mouse can activate object
8	Container	Can carry other objects (vehicles only)
16	Invisible	Creatures can not see it
32	Floatable	Maintains it's position relative to window, not world
64	Has boundaries	Responds to collision events and size
128	Suffers Gravity	Falls due to gravity

Values for BHVR Click

Value	Meaning
0	Clicks have no effect
1	Monostable: Clicks activate, further clicks have no effect until object is inactive again
3	Toggle: First click activates, second deactivates
4	Cycle: First click activate 1, second activate 2, third deactivate

Values for BHVR Touch

Value	Meaning
0	Creature can take no actions
1	Creature can activate 1
2	Creature can activate 2
4	Creature can deactivate
8	Creature can hit
64	Creature can eat

Values for MESG command

Value	Event
0	Activate 1
1	Activate 2
3	Hit
4	Pickup
5	Drop
8	Enter scope
10	Push left
11	Push right
12	Eat

Values for ACTV

Value	Meaning
0	Inactive
1	Active 1
2	Active 2
3	Hit
4	Eat

Values for MOVS

Value	Meaning	Description
0	Autonomous	Default – normal object in world
1	Mouse-driven	Simple Object connected to mouse
2	Floating	Object is in fixed place on screen
3	In Vehicle	Object is carried in a vehicle
4	Carried	Object is carried by a creature

Values for STM# Command

Value	Name
These are the names for the in-built stimuli. The actual chemical stimulation that a creature gets from these stims is genetically defined for each creature.	
0	"Disappointment"
1	"Pointer pats me"
2	"Creature pats me"
3	"Pointer slaps me"
4	"Creature slaps me"
5	"It is approaching"
6	"It is retreating"
7	"I bump into wall"
8	"Object comes into view"
9	"Unrecognised word"
10	"Heard user speak"
11	"Heard creature speak"
12	"I am quiescent (periodic)"
13	"I've Activated1"
14	"I've Activated2"
15	"I've Deactivated"
16	"I am approaching (periodic)"
17	"I have retreated"
18	"I have Got"
19	"I have Dropped"
20	"I've stated need"
21	"I am Resting (periodic)"
22	"I am sleeping (periodic)"
23	"I am travelling (periodic)"
24	"I've been pushed"
25	"I've been hit"
26	"I've eaten something"
27	"<spare action>"
28	"involuntary action 0"
29	"involuntary action 1"
30	"involuntary action 2"
31	"involuntary action 3"
32	"involuntary action 4"
33	"involuntary action 5"
34	"involuntary action 6"
35	"involuntary action 7"
36	"Approaching edge"
37	"Retreating from edge"
38	"Falling through air"
39	"Impact post fall"
40	“Pointer says yes”
41	“Creature says yes”
42	“Pointer says no”
43	“Creature says no”
44	“Aggression”
45	“Mate”

Default Word List

Value	Word
These are the words contained in the in-built word list – and their positions within that.	
0	'stay'
1	'push'
2	'pull'
3	'stop'
4	'come'
5	'run'
6	'get'
7	'drop'
8	'why'
9	'rest'
10	'left'
11	'right'
12	'eat'
13	'hit'
14	' '
15	' '
16	'me'
17	'hand'
18	'button'
19	'nature'
20	'plant'
21	'egg'
22	'food'
23	'drink'
24	'dispensor'
25	'implement'
26	'cliff edge'
27	'detritus'
28	'medicine'
29	'toy'
30	'weather'
31	'badplant'
32	'nest'
33	'badbug'
34	'bug'
35	'badcritter'
36	'critter'
37	'seed'
38	'leaf'
39	'root'
40	'flower'
41	'fruit'
42	'mover'
43	'lift'
44	'computer'
45	'mediabox'
46	'message'
47	'leftright'
48	'incubator'
49	'teleporter'
50	' '
51	'machine'
52	'Norn'
53	'Grendel'
54	'Ettin'
55	'Shee'
56	'pain'
57	'sad'
58	'hungry'
59	'cold'
60	'hot'
61	'tired'
62	'sleepy'
63	'lonely'
64	'crowded'
65	'scared'
66	'bored'
67	'angry'
68	'friendly'
69	'hurt'
70	'choking'
71	'thirsty'
72	'stressed'
73	'yes'
74	'no'
75	'look'
76	'what'
77	'very'
78	'really'
79	'seriously'
80	'intensely'
81	'extremely'

Chemical List

Value	Name
These are the numbers for the complete chemical list, blank spaces indicate a chemical slot that has not been used yet.	
1	Pain
2	Need for Pleasure
3	Hunger
4	Coldness
5	Hotness
6	Tiredness
7	Sleepiness
8	Loneliness
9	Crowded
10	Fear
11	Boredom
12	Anger
13	Sex Drive
14	Injury
15	Suffocation
16	Thirst
17	Stress
18	Pain Increase
19	Need for Pleasure Increase
20	Hunger Increase
21	Coldness Increase
22	Hotness Increase
23	Tiredness Increase
24	Sleepiness Increase
25	Loneliness Increase
26	Crowded Increase
27	Fear Increase
28	Boredom Increase
29	Anger Increase
30	Sex Drive Increase
31	Injury Increase
32	Suffocation Increase
33	Thirst Increase
34	Stress Increase
35	Pain Decrease (Endorphin)
36	Need for Pleasure Decrease
37	Hunger Decrease (Saccharin)
38	Coldness Decrease
39	Hotness Decrease
40	Tiredness Decrease
41	Sleepiness Decrease
42	Loneliness Decrease
43	Crowded Decrease
44	Fear Decrease
45	Boredom Decrease
46	Anger Decrease
47	Sex Drive Decrease
48	Injury Decrease
49	Suffocation Decrease
50	Thirst Decrease
51	Stress Decrease
52	Reward
53	Punishment
54	Reinforcement
55	ConASH
56	DecASH1
57	Reward Echo
58	Punish Echo
59	DecASH2

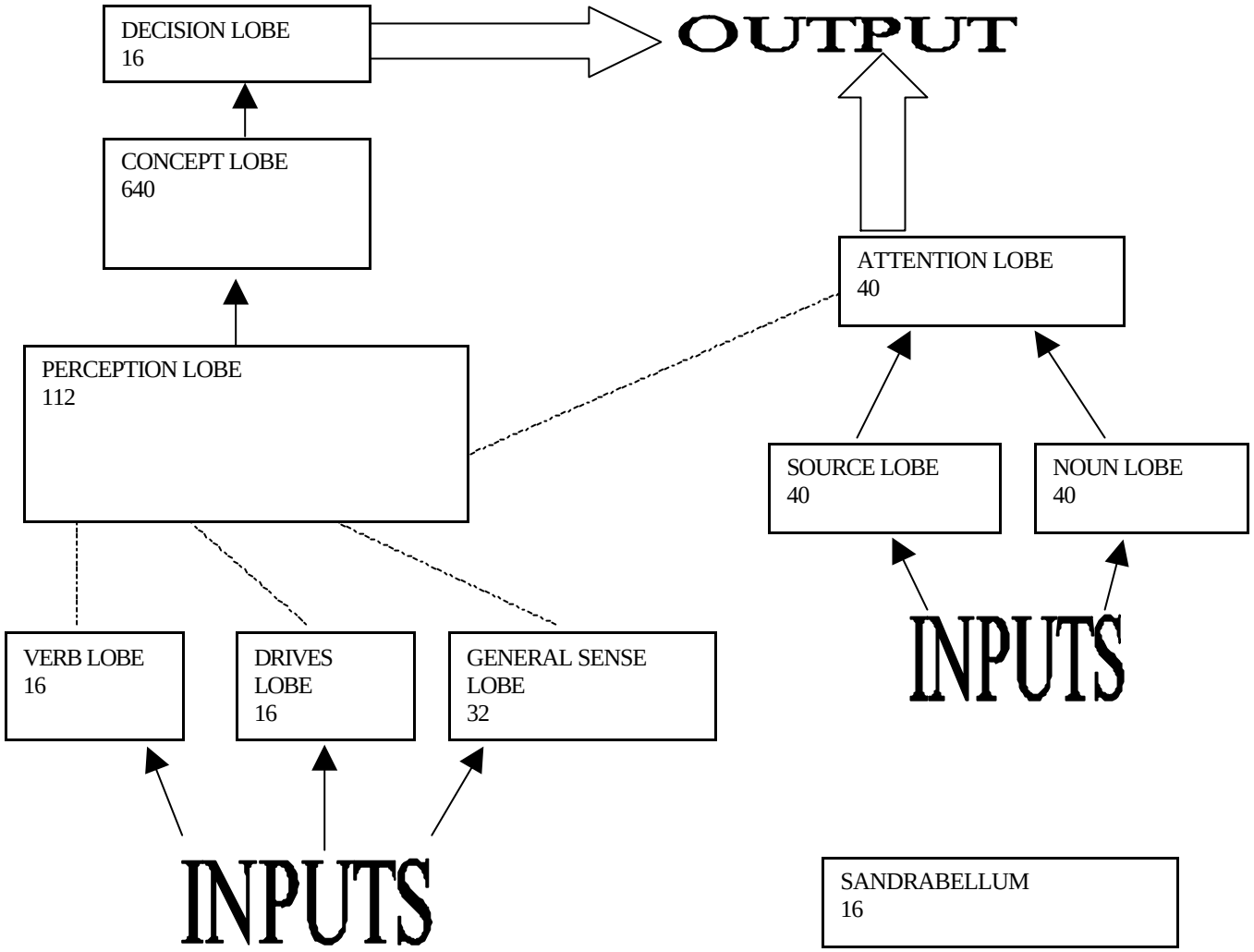
60	
61	
62	
63	
64	
65	
66	
67	
68	Lactate
69	Pyruvate
70	Glucose
71	Fatty Acid
72	Glycogen
73	Starch
74	Fat
75	Adipose Tissue
76	Ageing
77	Muscle Tissue
78	Triglyceride
79	Protein
80	Amino Acid
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	Carbon Dioxide
91	
92	Urea
93	Ammonia
94	
95	Oxygen
96	Air
97	Water
98	Energy
99	ATP
100	ADP
101	
102	
103	
104	Bilin
105	Oestrogen
106	Testosterone
107	Gonadotrophin
108	Progesterone
109	Inhibin
110	LH
111	FSH
112	Steroidone
113	Cholesterol
114	Arousal Potential
115	Mating Pheramone
116	Species Pheramone
117	Parent Pheramone
118	Child Pheramone
119	Sibling Pheramone
120	Opposite Sex Pheramone
121	Norn Smell
122	grendel smell
123	ettin smell
124	
125	

126	
127	
128	
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	Heavy Metals
141	Cyanide
142	
143	Belladonna
144	Geddonase
145	Glycotoxin
146	Fullness
147	
148	
149	
150	Vitamin E
151	Vitamin C
152	Bile Acid
153	Insulin Shots
154	Glycogen Synthetase
155	Dehydrogenase
156	Prostaglandin
157	EDTA
158	Sodium thiosulphite
159	Arnica
160	
161	
162	
163	
164	
165	
166	
167	
168	Tyrosine
169	Triptophan
170	Alcohol
171	Dancing
172	Adrenaline
173	Hexokinase
174	Activase
175	Turnase
176	Collapsase
177	downatrophin
178	upatrophin
179	
180	
181	
182	
183	
184	
185	
186	
187	
188	
189	

190	
191	
192	
193	
194	
195	
196	
197	
198	
199	
200	
201	
202	
203	
204	
205	
206	
207	
208	
209	
210	
211	
212	
213	
217	
218	
219	
220	
221	
222	
223	
224	
225	
226	
227	
228	
229	
230	
231	
232	Histamine A
233	Histamine B
234	Sleep toxin
235	Fever toxin
236	unknown toxin
237	unknown toxin
238	unknown toxin
239	unknown toxin
240	Antibody 0
241	Antibody 1
242	Antibody 2
243	Antibody 3
244	Antibody 4
245	Antibody 5
246	Antibody 6
247	Antibody 7
248	Antigen 0
249	Antigen 1
250	Antigen 2
251	Antigen 3
252	Antigen 4
253	Antigen 5
254	Antigen 6
255	Antigen 7

Brain Info

Brain wiring



Lobe Description

Name	**Description**
These are the lobes in a generation 1 creature	
PERCEPTION	Inputs into concept space and is a combination of inputs from other lobes. This lobe must be lobe 0
DRIVE	Monitors drive status – contributes to decision making
SOURCE	Who stimulus came from and how strongly – contributes to attention shift
VERB	Detects verbs spoken – contributes to decision making
NOUN	Detects nouns spoken – contributes to attention shift
SENSE	General sensory inputs about the environment, myself and IT
DECISION	What to do
ATTENTION	What to attend to or seek out
SANDRABELLUM	Completely detached. Acts as a mass to which chemo-regulators can attach.

Action numbers/offsets into DECISION_LOBE & VERB_LOBE

Neurone Number	Action
The neurones for these two lobes correspond with the actions that a creature can take.	
0	Quiescent, either introspective or extraspective depending on whether there is an _IT_ the creature is looking at.
1	Activate 1
2	Activate 2
3	Deactivate
4	Approach
5	Retreat
6	Get
7	Drop
8	Express Need
9	Rest
10	Travel West
11	Travel East
12	Eat
13	Hit

Drive numbers/offsets into DRIVE_LOBE

Neurone Number	Drive
The neurones for this lobe correspond with the possible drives a creature can feel.	
0	PAIN
1	NEEDFORPLEASURE
2	HUNGER
3	COLDNESS
4	HOTNESS
5	TIREDNESS
6	SLEEPINESS
7	LONELINESS
8	OVERCROWDEDNESS
9	FEAR
10	BOREDOM
11	ANGER
12	SEXDRIVE,
13	INJURY
14	SUFFOCATION
15	THIRST
16	STRESS

Offsets into SENSE_LOBE

Neurone Number	Meaning	Description
These neurones are the perceptions that a creature will use to form a view of the world, and the objects within it. (i.e. General sensory data about me, my environment and IT)		
0	PAT	I've been patted on head
1	SLAP	I've been slapped on bum
2	BUMPED	I've hit a wall
3	NEARWALL	I am this near to a wall
4	INVEHICLE	I am inside a vehicle
5	USERSPOKE	pointer has spoken (command)
6	CREATURESPOKE	another creature has spoken
7	MYKINDSPOKE	a creature of my species spoke
8	SOUND	an obj has emitted a sound
9	VISION	an obj has made a visible change
10	APPROACHING	IT is approaching
11	RETREATING	IT is moving away
12	ITNEARNESS	IT is this close to me
13	ITISACTIVE	IT is active1 or 2
14	ITISOBJECT	IT is a non-creature
15	ITISCREATURE	IT is a creature
16	ITISMYSIBLING	IT is my sister/brother
17	ITISMYPARENT	IT is my mother/father
18	ITISMYCHILD	IT is my son/daughter
19	ITISOPPOSITESEX	IT is a member of opposite sex and my genus
20	PUSHED	IT has pushed me
21	HIT	IT has hit me
22	SPARE4	placeholder
23	SPARE5	placeholder
24	SPARE6	placeholder
25	SPARE7	placeholder
26	SPARE8	placeholder
27	SPARE9	placeholder
28	APPROACHEDGE	I am approaching an edge
29	RETREATEDGE	I am retreating from an edge
30	FALLING	I am falling
31	IMPACT	I have hit the floor

Offsets into NOUN_LOBE, SOURCE & ATTENTION_LOBE

Neurone Number	Corresponds with…
These neurone numbers correspond to the 40 categories of object a creature can learn about – these categories are from the word list	
0	SELF - 'me'
1	SYSTEM - 'hand'
2	CALL BUTTON - 'button'
3	NATURE - 'nature'
4	GOOD PLANT - 'plant'
5	CREATURE EGG - 'egg'
6	PROCESSED FOOD - 'food'
7	DRINKS AND FOUNTAINS - 'drink'
8	FOOD DISPENSER - 'dispenser'
9	IMPLEMENTS AND DEVICES - 'implement'
10	CLIFF EDGE - 'cliff edge'
11	ROTTEN FOOD - 'detritus'
12	MEDICINAL POTIONS - 'medicine'
13	TOYS - 'toy'
14	CLOUDS AND PRECIPITATION - 'weather'
15	BAD PLANT - 'badplant'
16	ANIMAL NEST - 'nest'
17	BAD BUG - 'badbug'
18	BUG - 'bug'

19	BAD CRITTER - 'badcritter'
20	CRITTER - 'critter'
21	SEEDS - 'seed'
22	LEAVES - 'leaf'
23	ROOT VEGETABLES - 'root'
24	FLOWERS - 'flower'
25	FRUIT - 'fruit'
26	VEHICLES - 'mover'
27	LIFTS AND UP/DOWN MOVERS - 'lift'
28	TEACHING DEVICES - 'computer'
29	VISUAL AIDS - 'mediabox'
30	POP-UP MESSAGE WINDOW - 'message'
31	LEFT/RIGHT MOVERS - 'leftright'
32	INCUBATORS AND EGG HATCHERS - 'incubator'
33	TELEPORTATION DEVICES - 'teleporter'
34	' '
35	HEAVY MACHINERY - 'machine'
36	NORNS - 'Norn'
37	GRENDELS - 'Grendel'
38	ETTINS - 'Ettin'
39	SHEE - 'Shee'

Values for LCUS

Receptor Loci (type = 0)

'Organ'	**Tissue**	**Locus ID**	**Description**
The use of the term 'Organ' here is confusing – here it is used to determine between brain and body, and has nothing to do with organs defined in the genome. The loci for genetically defined organs are not accessible through the LCUS command			
0 Brain	Use lobe number	0 LOCUS_THRESHOLD	Cell threshold
		1 LOCUS_LEAKAGE	Cell leakage rate
		2 LOCUS_RESTSTATE	Cell rest state
		3 LOCUS_RELAXSUS0	Dentype 0 susceptibility decay rate
		4 LOCUS_RELAXSTW0	Dentype 0 Short-term synapse decay rate
		5 LOCUS_LTWRATE0	Dentype 0 synaptic consolidation rate
		6 LOCUS_GAINSTRENGTH0	Dentype 0 linear strength gain rate
		7 LOCUS_LOSESTRENGTH0	Dentype 0 linear strength loss rate
		8 LOCUS_RELAXSUS1	Dentype 1 susceptibility decay rate
		9 LOCUS_RELAXSTW1	Dentype 1 Short-term synapse decay rate
		10 LOCUS_LTWRATE1	Dentype 1 synaptic consolidation rate
		11 LOCUS_GAINSTRENGTH1	Dentype 1 linear strength gain rate
		12 LOCUS_LOSESTRENGTH1	Dentype 1 linear strength loss rate
		13 LOCUS_CHEM0	General-purpose loci for use by svrules.
		14 LOCUS_CHEM1	
		15 LOCUS_CHEM2	
		16 LOCUS_CHEM3	
		17 LOCUS_CHEM4	
		18 LOCUS_CHEM5	
		19 LOCUS_STATE0	State of first cell in lobe – other cells follow on. I.e. LOCUS_STATE0 + cellnum
1 Creature	0 Somatic	0 LOC_AGE0	If on and currently AGE_BABY then become AGE_CHILD.
		1 LOC_AGE1	If on and currently AGE_CHILD, become AGE_ADOLESCENT
		2 LOC_AGE2	
		3 LOC_AGE3	
		4 LOC_AGE4	
		5 LOC_AGE5	AGE_ADULT -> AGE_SENILE
		6 LOC_AGE6	If on, die immediately of old age.
	1 Circulatory	0 LOC_FLOATING0	General purpose emitter and receptor loci for doing more complex chemical regulation.
		1 LOC_FLOATING1	
		2 LOC_FLOATING2	
		3 LOC_FLOATING3	

		4 LOC_FLOATING4	
		5 LOC_FLOATING5	
		6 LOC_FLOATING6	
		7 LOC_FLOATING7	
	2 Reproductive	0 LOC_OVULATE	If low, remove any egg/sperm from gamete; if high add one.
		1 LOC_RECEPTIVE	If >0, female is receptive to incoming sperm and will conceive.
		2 LOC_CHANCEOFMUTATION	
		3 LOC_DEGREEOFMUTATION	
	3 Immune	0 LOC_DIE	If on, creature dies (ill health, poison, starvation…)
	4 Sensorimotor	0 LOC_INVOLUNTARY0	Trigger involuntary actions (fits, flinches etc)
		….	
		7 LOC_INVOLUNTARY7	
		8 LOC_GAIT0	Trigger various walking gaits (0=default, usually no need for a receptor here).
		…	
		24 LOC_GAIT16	
	5 Drives	0 LOC_DRIVE0	Drive levels (both receptors and emitters)
		…	
		16 LOC_DRIVE16	

Emitter loci (type = 1)

'Organ'	Tissue	Locus ID	Description
0 Brain	Use lobe number	0 LOCUS_ACTIVITY	Current amount of cell activity in lobe (# cells firing)
		1 LOCUS_NUMLOOSE0	Number of loose dens/cell in lobe (type 0 dens)
		2 LOCUS_NUMLOOSE1	Ditto for type 1 dens
		3 LOCUS_OUTPUT0	Output for first cell in lobe. Get other cells in lobe using LOCUS_OUTPUT0 + cellnumber as an ID
		4 Other cell outputs follow on from here – one locus per cell.	
1 Creature	0 Somatic	0 LOC_MUSCLES	How much energy has been expanded on movement this tick
	1 Circulatory	0 LOC_FLOATING0	General purpose emitter and receptor loci for doing more complex chemical regulation.
		1 LOC_FLOATING1	
		2 LOC_FLOATING2	
		3 LOC_FLOATING3	
		4 LOC_FLOATING4	
		5 LOC_FLOATING5	
		6 LOC_FLOATING6	
		7 LOC_FLOATING7	
	2 Reproductive	0 LOC_FERTILE	255 if a male has sperm or a female has an egg available
		1 LOC_PREGNANT	255 if female has both egg and sperm so is pregnant
	3 Immune	0 LOC_DEAD	>0 if creature is dead (allows post-mortem chemistry)

	4 Sensorimotor	0 LOC_CONST	Constant 255 (for regular emitters)
		1 LOC_ASLEEP	255 if asleep, else 0
		2 LOC_COLDNESS	How far air temp is below blood temp
		3 LOC_HOTNESS	How far air temp is above blood temp
		4 LOC_LIGHTLEVEL	How bright the sky is (eg control sleepiness)
		5 LOC_CROWDEDNESS	How many and how close others of your kind are.
		6 LOC_RADIATION	
		7 LOC_TIMEOFDAY	
		8 LOC_SEASON	
		9 LOC_AIRQUALITY	
		10 LOC_UPSLOPE	How steep is the slope I'm facing?
		11 LOC_DOWNSLOPE	
		12 LOC_HEADWIND	Speed of wind coming toward me.
		13 LOC_TAILWIND	Speed of wind coming from behind me.
	5 Drives	0 LOC_DRIVE0	Drive levels (both receptors and emitters)
		…	
		16 LOC_DRIVE16	

Index