Editing / Creating a task

If you want to edit or create a task this can now no longer be done here. Please see the section below on Task Editor.

Tasks are no longer shown on the wiki please log in to JiC and then go to Test task list. From here you will be able to view, edit and run tasks.

If you feel a task is ready to be added to the system, set the status as appropriate in the editor.

Running Tasks

When running a task you have two ways:

- Test - This shows more debugging information at the bottom of the screen and ignores the effect of denial and chastity commands.
- Run - Runs the task as if it was a validated random task, no debug info and chastity lockups etc need to be observed.

Note both ways of running the test tasks will not affect mood levels - if you need increase Jennifers mood do a random task.

Task Script

The scripting language is relatively simple and hopefully easy to understand from reading what the commands do.

However a couple of things that are worth highlighting before you jump in:

- NEVER use double quotes (“)
- When assigning a variable a string value or using a string value in a condition use single quotes.

Lets start by looking at a simple example.

Simple Example

```
;This is a comment line. It doesn't do anything, but makes this code easier
to read!
show:I'm going to be nice and give you a chance to cum today. First, get
naked and fetch a pair of handcuffs.
wait:60
cam:1
button:Naked and ready, Mistress
show:Good, let's get started. Play with yourself, get yourself nice and
excited for me.
sound:bell.wav
```

```
wait:30
```

A-Z of Script commands

Recently added commands

savetemp1,savegtemp1,loadtemp1,loadgtemp1,loadtemp2,loadtemp3,savetemp2,savetemp3

Except for comment lines all commands are on a new line with a colon between the command and the parameter.

Command	Parameter	Description
asktext	prompt text	Text input box - result saved in LASTASK
asknumber	prompt text	number input box. that keeps asking until it gets a valid number (allows a decimal point). - result saved in LASTASK
blackmail	title:value	Records a piece of blackmail information.
button	Button Text	Displays a button with the given Button Text. Waits until the button is clicked.
cam	*Not used*	Take picture with camera. Pictures stay permanent, but are not put into the blackmail library
cam	blackmail	If you add blackmail as the parameter, it will also show the picture in the message center and add it to Jennifer's “blackmail gallery” she has on the user.Please don't use this for every cam command! I expect it to really be only used at most once per task or the mods will be flooded with images.
cam	broadcast	Picture gets displayed to all online users for 60 secs
cam	temp	Pictures get deleted after 4 weeks
setupdiff	*Not used*	Take reference image for camdiff
camdiff	*Not used*	Compare differences with setupdiff (diff level saved to TEMP1)
choice	option1,option2….	Create button for each option selected button saved in LASTASK
image	image file name	Display image file e.g. lib_image/02.jpg or img/1452733840.dat - You can upload an image file in the message system and right click on it to see the filename. Don't include the domain just from img on-wards e.g. image:img/1452733840.dat
kimage	0 or 1	Display a key compromising image of user (1) or picture of partner/crush (0)
log	Text to write	Write text as if user put in message system
nexttask	Name of task to do next time ask for random	*Note this does not work in test tasks*
dmsg	Hours of delay	Delayed message. e.g dmsg:1:Natasha:Do this now!
dtask	Hours of delay	Delayed task. e.g task:1:Natasha:Choices (“choices” is the name of the task)
settemp1	value to save to TEMP1	can also be 2, 3, 4 all the way to 9
savetemp1	Variable name	Store value of TEMP1 to user store, can also be 2,3

Command	Parameter	Description
loadtemp1	Variable name	Load value from user store to TEMP1 (value is that at time script starts) can also be 2,3
savegtemp1	Variable name	Store value of TEMP1 to global store
loadgtemp1	Variable name	Load value from global store to TEMP1 (value is that at time script starts)
assign	Variable name : Variable value	Save the value of a variable to a variable with the given name. e.g.: assign:saved_full_name:'Elisabeth Jane' .With this exemple, you can then retrieve the value of the variable with $saved_full_name . WARNING : Only the first occurrence will be replaced in a line of code.
setmood	should be current mood minus what you want to deduct	e.g. setmood:(MOOD-1) See example for more.
setchastity	value to set [days or hours]	e.g. setchastity:2 days
setdenial	value to set [days or hours]	e.g. setdenial:2 hours
addchastity	value to add to existing time [days or hours]	e.g. addchastity:2 days
adddenial	value to add to existing time [days or hours]	e.g. adddenial:2 hours
broadcast	msg text	Sends msg text to all currently online users for 1 min
show	Display Text	
showbar	value 0 to 100	
hidebar	NA	Hide showbar
joingame	game_name	Your are joined into a team game where all text and images are shared. It is a good idea to do “log:games stop” before this task to ensure the user is exited other games first.
goto	label_name	This will jump the script to the label name defined
label	label_name	This will define a point for the Goto function to jump to
sound	Sound file to play	available sounds: 20slowswats.wav, bell.wav, cane.wav, metronome.wav, shortwhip.wav, swat.wav, tom.wav, whip.wav
speak	text to speak	Use the voice of your current Mistress/Master. Will also work for “speakjen”,”speaknat“ and “speakjames”, using the specific voice. Keep in mind that you might have to spell it as you want it pronounced in some cases, such as “cum” = “come”
startgroup	number in group	add user into a group game for this task, once number reached starts new group - this is done when task loads so not conditional on if's etc
trigger	device:(delay in seconds)	trigger remote device e.g. trigger:vibe:12
upload	*Not used*	Allow user to upload an image file. To use it, explain what you are wanting the user to upload with a “show:” command and then use it to add a file selection, they can then upload any image their device can open. The script will pause until a valid image is uploaded. This can be used in combination with “dmsg:” and “dtask:” to get someone to go away and do something before reporting back with a picture to upload.
wait	number of seconds to wait	

Command	Parameter	Description
waitm	number of milliseconds to wait	
wbutton	seconds till auto click:button text	e.g. wbutton:10:press me = display “press me” and move on if not pressed in 10 seconds - LASTASK = 0 if clicked or 1 if timeout
web	http://www.google.com	open page in new window
finish	*Not used*	End task and display “Return to message view” button

Variables

To set temp values: settemp1:(expression) - e.g. settemp1:'Hello'

When using variables be careful with non numerical values e.g.

```
if:(TEMP1 == 2)
```

is fine if TEMP1 contains a number but if not you should use quotes for both sides e.g.

```
if:('TEMP1' == 'two')
```

Variables can be used in all parameters and are simply swapped for their value

Variable	Possible Values	Description
TEMP1 to TEMP9	anything	Store whatever you like in TEMP1, TEMP2 up to TEMP9
MOOD	0-100	The current mood from the Master or Mistress of the slave
LASTASK	anything	Contains the value of of last asktext
FEMALE	true,false	Contains true if slave is a Female, otherwise false
TRANS	true,false	Contains true if slave is Trans, otherwise false
DNAME	Jennifer,James,Natasha	Contains the name of the selected Master/Mistress
DTITLE	Master or Mistress	Contains the title of the selected Master/Mistress
SNAME	anything	Contains the name the slave
PNAME	anything	Contains the name the slave crush or partner
P_JIC	true,false	True if in crush or partner knows about JiC
TIMENOW	(current time)	Current time in milliseconds (handy for timing)
DENIED	true,false	True if not allowed to cum, otherwise false
INCHASTITY	true,false	True if in chastity, otherwise false
HARDCORE	true,false	True if in hardcoremode, otherwise false
BRIGHTNESS		The brightness value from the last image (setdiff or cam)
CONTRAST		The contrast value from the last image (setdiff or cam)
EN_NOISE	true,false	True if need to be careful about noise, otherwise false
EN_HOME	own, partner, kids, parent, friends	Who the slave does live with

Variable	Possible Values	Description
S_LIKE_W	0,1,2	Fantasy Partner preference for Woman: 0 (Not my thing), 1 (Somewhat attractive), 2 (rock my world)
S_LIKE_M	0,1,2	Fantasy Partner preference for Men: 0 (Not my thing), 1 (Somewhat attractive), 2 (rock my world)
S_LIKE_T	0,1,2	Fantasy Partner preference for Trans: 0 (Not my thing), 1 (Somewhat attractive), 2 (rock my world)

Flow control

Warning this is very simplistic - no nested loops!, however nested if blocks should now be fine If you are using if,loop or while blocks it is easier to read if you indent the commands in between with 2 spaces 😉

```
if:(condition)
fi:

if:(condition)
  ; if condition is truish
else:
  ; otherwise
fi:

loop:5 - loops the var LOOPVAR 1..5
endloop:

settemp1:1
while:(TEMP1 < 4)
  settemp1:(TEMP1+1)
endwhile:

;define a function - can be anywhere in task
function:ButtonFunction
  button:TEMP1
end:

;call the function as many times as you like
settemp1:'button1'
call:ButtonFunction
settemp1:'button2'
call:ButtonFunction
```

Require

Require options are as follows (1st 3 are remote devices - last one only allowed to run if sub does not have cum restrictions) :

```
restim rshock rlock chastity buttplug candle crop dildo estim gag handcuffs
```

```
humbler nippleclamps nochaste cancum
```

*note: chastity here is that the "user is currently locked in chastity", if you need to require they have a device, setting "Chastity" in **limits** also checks they have a device.*

You can also check the available toys at runtime with variables. Even if the sub does not have one of the following toy (value at False), the task could be run as long as the script plays around it, and the toy does not have to be a requirement to run the task. On the opposite, it becomes a 'bonus' for the task.

```
T_BUTTPLUG
T_CANDLE
T_CROP
T_DILDO
T_ESTIM
T_GAG
T_HANDCUFFS
T_HUMBLER
T_NIPPLECLAMPS
```

e.g.:

```
if:(T_GAG)
```

Limits

Limit selections are as follows (in the same order as shown in options):

```
Anal ATM Bondage Chastity Cum Face Financial Online Outdoor Exhibitionism
Orgasm Pain-G Pain-N PInfo Sissy Shaving Watersport
```

You can also retrieve the values of the different limits at runtime, with -1 for the No (HardLimit) and up to 5.

```
L_ANAL
L_ATM
L_BONDAGE
L_CHASTITY
L_CH_TYPICAL - typical time of chastity in days (0 = just a few hours)
L_CUM
L_FACE - Where 1+ can show face to system, 2+ people doing reviews, 3+ other
people in JIC, 5 online
 - (e.g. use  if:(L_FACE < 2) peerreview:false;)
L_FINDOM
L_ONLINE - Tasks exposing yourself online (wearing mask or hiding identity)
L_OUTDOOR
L_EXHIB
L_DENIAL
L_PAIN_G
```

```
L_PAIN_N
L_INFO
L_PERSONALINFO (same as L_INFO)
L_SISSY
L_SHAVE
L_WATERSPORT
L_PERSONALINFO
```

e.g.:

```
if:(L_SISSY > 3)
```

A typical use of this would be to change the difficulty of a part of the task, or bypass this part completely, without forbidding the whole task from being run by the sub just because of this limit. Remember: if the mood is low enough the sub can be pushed more (soft limits).

Libraries

Based on the current test tasks we will soon have a selection of libraries you can use to save rewriting common tasks. The following is an example of how they may be used.

```
;Use the Cards library
use:cards
;Draw a card (value stored in TEMP1)
call:GetCard
;Show result
show:You got the card <img src='cards/TEMP1.bmp'>, I hope that's better than
mine for your sake!
;Extract value of card stored in TEMP1 into TEMP2
call:CardValue
if:(TEMP2 < 7)
  show:Oh bad luck
fi:
```

Currently we have 2 libraries:

cards

function:GetCard - draw a random card and store in TEMP1

function:CardValue - save value of card in TEMP1 to TEMP2

video

function:TestCamera - keeps checking till camera working. UPDATE: The TestCamera is not needed as the video check is done automatically by the JiC messenger before allowing the running of a task.

Task Editor

Watch this space for a new way to edit/create tasks...

From the Test task list you can now edit or create tasks using the new Editor. For more information see Editor Guide

Messages | New Task | Task Images | Wiki

Edit Task: button test

Top section include title, sexual identification, Task requirements and limits that will be affected in this task

Title: button test
Sex: ☑ Male ☑ Female ☑ Trans (Female T Male T Trans T)
Require: ☐ In Chastity ☐ Not in Chastity ☐ Can Cum ☐ Remote estim ☐ Remote shock ☐ Remote lock ☐ Buttplug ☐ Candle ☐ Crop ☐ Dildo ☐ E-Stim ☐ Gag ☐ Handcuffs ☐ Humbler ☐ Nippleclamps ( )
Limits: ☐ Anal ☐ Bondage ☐ Chastity ☐ Cum ☐ Financial ☐ Outdoor ☐ Exhibitionism ☐ Orgasm ☐ Pain-G ☐ Pain-N ☐ Sissy ☐ Shaving ☐ Watersport ( )

Todo List
1. Better help when editing commands e.g. help for current selected command
2. Save with version history
3. Basic checking of evaluation for if statements
4. Preview in "image:" command + Maybe allow a selection of default images?

Drag lines to reorder or click to change.

Middle section is the actual action form the script. You can add/remove, drag and drop lines with functions around. And the task will do whats on each line starting from line 1

Start

```
1  ;
2  Show Be honest now, did you cum?
3  Choice no,yes
4  ;
5  Wait 3
6  ;
7  If (LASTASK==1)
8    Show Thats just too bad, you obviously need to be more needy first SNAME.
9    Button Yes DTITLE
10   SetDenial 1 hour
11 FI
12 ;
13 If (LASTASK==2)
14   Show Good, now show your proof to the camera.
15   Button Look DTITLE
16 FI
17 ;
18 Show We are done here SNAME.
19 Button Thank you DTITLE.
20 ;
```

Finish

21 New Command (drag me to where needed)
22 Delete Command (drag command to delete here)

Status

Comments for change log:
Save

In the last section you will find the ability to make a change-log, especially useful if you are working with others to create a task.

Also the text box allows external editors and last you can see the history of what what changed, when and by who.

Advanced

(To allow copy and paste to externals tasks)

```
;
show: Be honest now, did you cum?
choice:no,yes
;
wait:3
;
if:(LASTASK==1)
  show: Thats just too bad, you obviously need to be more needy first SNAME.
  button: Yes DTITLE
  setdenial: 1 hour
```

Check Advanced

History

No history available for this task

Example Code Segments

Random Loop

```
;Sets TEMP1 to any number from 1 to 10
settemp1:Math.floor((Math.random() * 10) + 1)
loop:TEMP1
  sound:bell.wav
  wait:3
endloop:
```

Disable peer review based on face limit (given that this task takes face images)

if:(L_FACE < 2)

```
peerreview:false
```

fi:

Take Picture Randomly

```
;This fragment can replace simple wait+cam fragments to make them less
predictable
;Change both '60's to be the delay before moving to the next section
settemp1:Math.floor((Math.random() * 60) + 1)
wait:TEMP1
cam:1
wait:60-TEMP1
```

Differentiate Physical Gender

```
;but don't forget TRANS
if:(FEMALE)
  settemp1:'girl'
fi:
if:(!FEMALE)
  settemp1:'boy'
fi:
show:Good TEMP1! :)
```

Identify as and Like

```
;Identify mostly as 1= Man 5 = Woman
if:(S_IDENTIFY > 2)
  settemp1:'girl'
fi
if:(!FEMALE)
  settemp1:'boy'
fi:
if:(S_LIKE_W > 1)
  settemp2:'like'
fi:
if:(S_LIKE_W == 1)
  settemp2:'dont like'
fi:
show:Good TEMP1, I see you TEMP2 girls!
; How much like women 1 = no 2 = a bit 3 = a lot
; Also can use S_LIKE_M and S_LIKE_T
```

Button times

```
show: Can you click yes in less than 10 seconds?
settemp1:TIMENOW
wbutton:10:Yes
settemp1:TIMENOW-TEMP1
if:(LASTASK==0)
  show: Well done. Clicked in TEMP1 msec
  button: Ok
fi:
if:(LASTASK>0)
  show: Too slow.
  button: Sorry
fi:
```

Get and display a random card

```
use:cards
;Draw a card (Card stored in TEMP1 value stored in TEMP2)
call:GetCard
call:CardValue
show:You got the card <img src='cards/TEMP1.bmp'>, with a value of TEMP2
```

Get results based on random 1-3 number gen

```
button:spin the wheel of fortune
;Sets TEMP1 to any number from 1 to 3
settemp1:Math.floor((Math.random() * 3) + 1)

if:(TEMP1 == 1)
```

```
  show:Lucky boy, you get to cum today.
  button:Thank you DTITLE
fi:
if:(TEMP1 == 2)
  log: set denial 1 days
  show:Denied for 24 hours, what a shame.
  button:Thank you DTITLE
fi:
if:(TEMP1 == 3)
  log: set denial 2 days
  show:Denied for 48 hours, what a shame.
  button:Thank you DTITLE
fi:
```

Delayed message and task assignment

```
;1 hour delay until Natasha says: "Do this bitch."
dmsg:1:Natasha:Do this bitch.
;
;1 hour delay before Natasha assigns the "choices" task.
dtask:1:Natasha:Choices
```

Mood adjustment

```
;Making sure that the current mood is show in the message
;center so it can be traced back if needed.
log:Current Mood = MOOD
;
;then setting the mood minus how much you want to deduct
;from the current mood. can only go down currently.
setmood:(MOOD-1)
```

Function usage and examples

AI Msg

Description: To be done

Code example:

```
To be done
```

AddChastity

Description: Addchastity adds time to the current chastity time. If units are not included in the

parameters, the default used is hours. Careful using this in test tasks!!!

Code example:

```
addchastity:12 hours
addchastity:12
```

AddDenial

Description: Adddenial adds time to the current chastity time. If units are not included in the parameters, the default used is hours. Careful using this in test tasks!!!

Code example:

```
adddenial:12 hours
```

AskText

Description: Text display and input box with the result saved in LASTASK variable. This replaces any other text displayed so do not use a preceding show command.

Code example:

```
asktext:What is your real name
show:Well LASTASK I think I will still call you slave
```

AskNumber

Description: Number input box with prompt text. That keeps asking until it gets a valid number (allows a decimal point)

Code example:

```
asknumber:Enter your height in cm
show:So you are LASTASK cm tall
```

Assign

Description: Allows user defined variable names. Parameters include the variable name and the value. If used in in a show or other similar command the variable name must be preceded with a $ sign.

Code example:

```
assign:myvar1:0
show:The number is $myvar1
```

Blackmail

Description: When you receive sensitive information, cataloging with the blackmail function allows it be be found easily by mods. *note* this has been depreciated for images - you should now use cam:blackmail instead. Remember if you are asking very sensitive information the task should also check for (L_INFO > 1) and also set peerreview:false **Code example:**

```
asknumber:How many sexual partners have you had?
blackmail:The slave had that amount of sexual partners:LASTASK
```

Button

Description: Shows a button on the screen, this is used usually to start an action like take an image or go tho the next message.

Code example:

```
button:Take image
```

Call

Description: Call a function within your script if you have written your own functions. Look at the section “Function & End” to find out how to write own functions

Code example:

```
call:yourFunctionName
```

Cam

Description: Commands the users camera to take a picture. The parameter temp is used for standard images, which are then deleted after sometime in order to save server space. Please use the temp parameter for most images. Using no parameter will keep the image permanently. The blackmail parameter alerts the mods to this image. Only use the blackmail parameter for the most humiliating images.

Code example:

```
cam:
cam:temp
cam:blackmail
```

CamDiff

Description: Camdiff detects the change in an image and report the difference as a number using

TEMP1. The setupdiff is recommended to set an initial image reference, but is not need for subsequent camdiff commands as each camdiff can use the previous camdiff as reference. The value (TEMP1) can then be used/evaluated. Shown below is the use of camdiff to detect movement.

Code example:

```
;TEMP1 is used by camdiff and setupdiff to log movement value
;TEMP2 is time in minutes
;TEMP3 is time in ms when task should be over or while ends
;TEMP4 is Movements detected
;
settemp2:1
settemp3:TEMP2 * 60000 + TIMENOW
settemp4:0
setupdiff:0
;
while:(TEMP3 > TIMENOW)
  wait: 2
  camdiff:0
  show:Movement value is TEMP1
  if:(TEMP1 < 2) && (TEMP1 > 1.5)
    speak: Steady
  fi:
  if:(TEMP1 < 2.5) && (TEMP1 > 2)
    speak: Careful slave
  fi:
  if:(TEMP1 > 2.5)
    cam:
    wait:1
    speak: You moved slave! Stand still!
    log: Movment detected. Slave should still be standing there, not moving.
    settemp4: TEMP4 + 1
    if:TEMP4 == 4
      speak:You have moved at least 4 times.  Task failed.
      ;sets TEMP3 to less than TIMENOW to end the while loop
      settemp3:TEMP3-(TEMP2 * 60000)
    fi:
  fi:
  ;
endwhile:
;
```

Choice

Description: The choice function prints buttons whit text which the slave has to chose one from. Then you can recive the result in LASTASK which contains the number of the choosed button. “Yes” would be 1 and “No” 2 in this example.

Code example:

```
choice:Yes,No
;then get the result with LASTASK and use it in a if.
```

Comment

Description: This is used to add comments, notes and details to the program/code which are not run or evaluated by the computer. It is used at the beginning of a program to list the details about the program including variables etc. Other users can also comment or recommend edits to the program using the comment command It can also add descriptions and explanation of complex equations

Code example:

```
;Buttplug_training by Scott2, updated 6/24/2019 by Scott2
;Notes- changed show/asktext to asktext, added logs
;Todo list-
;
;Variable table-
;use TEMP1 for reading and saving data
;use TEMP2 for girth of plug inserted today
;use TEMP3 for previous task run info, typically plug girth
;use TEMP4 for previous task run info, time plugged
;use TEMP5 for time plugged today
;
```

Delay Msg

Description: dmsg:x:y:z Delayed message *z*, to appear in the message center *x* hours from the chosen person *y*. For ref 1 day = 24 hours, 7 days = 168 hours *y* will normally be Jennifer, James or Natasha

Code example:

```
dmsg:24:Natasha:I hope you remembered to do a random task in the last day?
```

Delay Task

Description: dtask:x:y:z Delayed task, a link to *z* will appear in the message center *x* hours from the chosen person *y*. For ref 1 day = 24 hours, 7 days = 168 hours *y* will normally be Jennifer, James or Natasha

Code example:

```
show:Once you are back from the shops, you will be set another task to
complete
dtask:3:Natasha:back_from_shop
```

Function & End

Description: The function command is used to create a subroutine with the name(title) of the routine denoted by the parameter. The end of the function is denoted by the end command. The end command returns the program to the position of the last call command. The parameter box for the end command can be used to note the name of the function it is paired with. A function is run by issuing the call command.

Code example:

```
function:Booknexttraining
  show:Can you return in three days for another round of training?
  choice:Yes,No
  if:LASTASK == 1
    show:Excellent choice.  See you again in 3 days.
    dmsg:71:Jennifer:Time for more buttplug training!
    dtask:71:Jennifer:Buttplug_training
    button:Thank You DTITLE DNAME
  fi:
  if:LASTASK == 2
    show:Is someone afraid of having their ass ruined?  Remember, your
asshole belongs to DTITLE DNAME.
    button:Yes, DTITLE DNAME
  fi:
end:Booknexttraining
```

Goto

Description: The goto command is used to jump from one position is a program to another position (label). The use of goto commands is normally not good programming practice, but is sometimes needed.

Code example:

```
goto:finished
;code to be skipped
label:finished
```

HideBar

Description: To be done

Code example:

```
To be done
```

If& FI

Description: Used as the basic decision making processes. The if: command is follow by a statement and if the statement is true the code continues to execute the code in the if block. If the statement is false the if block is skipped pass the point of the fi: command.

Code example:

```
if:TEMP7==3
  show:Pose for your exposure picture. On all fours with ass to camera.
  log:Sissy will now pose, On all fours with ass to camera.
fi:
```

Image and Image2

Description: Changes the background image using an image from the JiC image folder. The available images are located here: https://play-link.com/jennifer/taskimages.php To use an image, provide the image folder/filename.

Image2 functions similar, but is displayed in an image window on the right side of the display, above the background image.

You can add/upload your own images to the taskimages folder. Please be sure your images have been properly sized and are of good quality, as deleting images or editing of uploaded images is not available.

Background images are fit-stretched to the width of the display window; therefore the ideal dimensions for background images is 1200(width) x 600(height). Since the text area uses the left half of the display it is useful to right justify the image.

Images for use using image2 are tall and narrow with the ideal dimensions of 300(width) x 600(height).

Code example:

```
image:taskimg/1_2.jpg
image2:taskimg/Jen_Cuff1.png
```

Join Game

Description: To be done

Code example:

```
To be done
```

KImage

Description: Kimage display an embarassing image which has been saved by the moderators.

Code example:

```
;display the image of crush
kimage:0

;display image of slave
kimage:1
```

Label

Description: Used to “label” a point in the code, typically used with goto command. See goto.

LoadGTemp1

Description: Load value from global store to TEMP1 (value is that at time script starts)

Code example:

```
To be done
```

LoadTemp1

Description: Loads a saved variable and places it in TEMP1. If no variable is found, TEMP1 is “null”. It is good practice to check for null.

Code example:

```
loadtemp1:times_run
if:'TEMP1' == ''
  settemp1:0
fi:
settemp3:TEMP1
```

Log

Description: Logs an whatever text you like. A log is used for the person who reviews a task. For example describe in a log message what the Slave should do in an image.

Code example:

```
log:In the next picture the slave must be naked.
```

Loop & EndLoop

Description: The command loop executes a subroutine (code between the loop and endloop commands) a defined number of times. The number of times the loop has run is traced by the variable LOOPVAR.

Code example:

```
loop:60
  settemp8:(8 * LOOPVAR)
  image:taskimg/wheelTEMP7.jpg
  waitm:TEMP8
  settemp7:(TEMP7+1)
  if:(TEMP7==13)
    settemp7:1
  fi:
endloop:
```

NextTask

Description: This function schedules a set task to be run the next time a task is selected from the random pool of tasks.

Code example:

```
nexttask:Task_name
```

PeerReview

Description: PeerReview can be at the start of a task allowing others to review a task. This should be done by default for most task, if a task contains very sensitive information it should be turned off.

Code example:

```
;turn peerreview on
peerreview:true
;turn peerreview off
peerreview:false
```

SaveGTemp1

Description: Store value of TEMP1 to global store

Code example:

```
To be done
```

SaveTemp1

Description: SaveTemp1 saves the current value of TEMP1 (number or text) to the variable name given. This can then be retrieved at a later date using loadtemp1

Code example:

```
asktext:Measure your dildo and enter it below in inches.
settemp1:LASTASK
savetemp1:dildolength
```

Set Mood

Description: The function Set Mood (setmood) is used to add or subtract mood points. It is best to leave this function for the adminastrators to add. If an important part of a task, only add (or activate) this function after beta testing is complete.

Code example:

```
setmood:(MOOD+1)
```

SetChastity

Description: This changes the current value for chastity to a new value. Careful using this in test mode!!!!!!!

Code example:

```
setchastity:12 hours
```

SetDenial

Description: This changes the current value for denial to a new value. Careful using this in test mode!!!

Code example:

```
setdenial:12 hours
```

SetTemp1 - SetTemp9

Description: Used to set the variables TEMP1-TEMP9. The variable TEMP1 is used to load and save variable and used by camdiff, so it is recommended TEMP1 not be used for data needed later in the task.

Code example:

```
settemp2:1
settemp3:TEMP2 * 60000 + TIMENOW
```

SetupDiff

Description: Used with camdiff. See camdiff.

Show

Description: Writes simple text to the screen. This is usually used to communicate with a slave. The display of the text is replaced by issue of an asktext, asknumber or another show command.

Code example:

```
show:Hello world
```

ShowBar

Description: The showbar command does not appear to function at this time 8/8/2019

Code example:

```
To be done
```

Sound

Description: Plays a sound file. Very useful to signal the tasking of a picture or the end of a wait period.

Code example:

```
wait:10
cam:
sound:bell.wav
wait:1
```

Speak

Description: Renders a voice to speak the text.

Code example:

```
speak:10 seconds left
```

SpeakJames

Description: Same as speak, use a voice for Master James

Code example:

```
speakjames:Hello
```

SpeakJen

Description: Same as speak, use a voice for Mistress Jennifer

Code example:

```
speakjen:Hello
```

SpeakNat

Description: Same as speak, but uses Russian rendering. Appears to speak numbers and none english words in Russian. Try writing numbers out in english text for better results. For example use “ten” instead of “10”

Code example:

```
speaknat:Hello
```

Startgroup

Description: To be done

Code example:

```
To be done
```

Task

Description: To be done

Code example:

```
To be done
```

Trigger

Description: To be done

Code example:

```
To be done
```

Use

Description: The use command loads a library containing functions. Functions can then be called from the library. See libraries section above.

Code example:

```
use:cards
call:GetCard
```

WButton

Description: This is a “wait button”. The button appears and is active for a given number of seconds. If clicked on the button proceeds as a normal button function. If the time elapses the button disappears and the program continues

Code example:

```
wbutton:10:Yes
```

Wait

Description: Wait is used for timing. If you have a wait command in your script the script stops for the amount of time you defined until the next step is triggered. The wait command needs an amount of seconds to work.

Code example:

```
show:Hello world
wait:10
show:now 10 seconds have passed.
```

WaitM

Description: Same as wait, but value is in milliseconds

Code example:

```
show:Hello world
wait:500
show:now 500 milliseconds have passed.
```

Web

Description: The command opens a web page in the default internet browser.

Code example:

```
web:https://www.faproulette.co/TEMP1
```

While & EndWhile

Description: Used to repeat a set of code until the condition (in the while line of code) is true.

Code example:

```
settemp1:10
while:TEMP1>0
  image:taskimg/strapon2.jpg
  show:You have TEMP1 seconds before I begin pegging you
  wait:1
  settemp1:(TEMP1 - 1)
endwhile:
;
```

From:
https://play-link.com/wiki/ - **PlayLink**

Permanent link:
https://play-link.com/wiki/doku.php?id=task_editor&rev=1565454551

Last update: **2019/08/10 18:29**