

Title:slave_task_market
Sex: Female T Male T Trans T
Require:
Limits:
assign:task1:'ball_busting_1;P;Pain-G;Getting the balls busted with a special device;'
assign:task2:'Bondage_showcase;PBN;Bondage;Demonstrating ones bondage equipment;'
assign:task3:'Bored;PBN;NotinChastity;Jennifer does some random things;'
assign:task4:'Bow_to_your_owner;PN;NotinChastity,Humbler,Pain-G;Bowling to their Mistress;'
assign:task5:'Brain_Train;P;;Stroking a dildo;'
assign:task6:'chastity_mercy_of_the_crowd;PBN;ChastityDevice;Temp expose a photo, asking others for chastity extension;'
assign:task7:'competitive_spanking_1;PBN;Crop,Pain-N;Trying to spank harder than the previous slave;'
assign:task8:'confession_1;PBN;;Confess something you did and get judged by the reviewer;'
assign:task9:'CornerTimeA;PBN;;Standing in a corner;'
assign:task10:'CornerTimeB;PBN;;Standing in a corner, but being able to trade time for pain;'
assign:task11:'crawlF;PBN;Bondage;Crawling while being in bondage;'
assign:task12:'crawlM;P;Bondage;Crawling while being in bondage;'
assign:task13:'deepthroat_training_1;PBN;Dildo;Practicing to deep throat a dildo;'
assign:task14:'detention;PBN;Crop,Pain-N;Being in detention with a strict teacher for some time;'
assign:task15:'Difficult_choices;PBN;;Having to choose between hard choices, at least one of them pushes the limits;'
assign:task16:'find_my_pussy_1;PBN;;Match girls to their pussies and get nipple pain for mistakes;'
assign:task17:'game_1;PBN;Nippleclamps,Pain-N;A reaction game with nipple pain for mistakes;'
assign:task18:'hand_writing_lines_nipples_clamped;PBN;Nippleclamps,Pain-N;Hand writing lines while the nipples are clamped;'
assign:task19:'housework;PBN;;Doing some cleaning or tidying up;'
assign:task20:'ice_torture_1;PBN;Pain-G,Pain-N;Runa has some fun with ice cubes;'
assign:task21:'interrogation_1;PBN;Nippleclamps,Pain-N;Interrogation with nipple clamps by Sister Evie;'
assign:task22:'jumping_jacks_cbt;P;NotinChastity,Pain-G;Jumping jacks with a weight attached to the balls;'
assign:task23:'karada;PBN;Bondage;Doing a karada bondage;'
assign:task24:'know_your_mistress_1;P;Nippleclamps,Pain-N;Identifying JiCs dominants by their feet;'
assign:task25:'lines;PBN;;Hand writing lines;'
assign:task26:'Measuring_up;PBN;NotinChastity;Measuring the slaves most interesting body parts;'
assign:task27:'mercy_of_the_crowd_2;PBN;Online;Begging to get a task assigned by somebody;'
assign:task28:'Nipple_clamp_torture;PBN;Nippleclamps,Pain-N;Cycles of

```
attaching and removing nipple clamps;'
assign:task29:'nipple_torture_1;PBN;Crop,Nippleclamps,Pain-N;Runa tortures
the nipples in various ways;'
assign:task30:'nipple_torture_2;PN;Nippleclamps,Pain-N;Spanking off nipple
clamps with a towel;'
assign:task31:'nipple_torture_3;PBN;Nippleclamps,Pain-N;Quickly attaching
and removing nipple clamps;'
assign:task32:'overstimulation_1;PBN;NotinChastity;Rubbing the cock head or
clit until it gets unpleasant;'
assign:task33:'pee_inside_or_outside;PBN;Watersport;Rubbing pee on the body
with a chance of having to drink it;'
assign:task34:'Position_Training;PBN;;Learning slave positions;'
assign:task35:'questions_1;PBN;NotinChastity;Testing the JiC knowledge of
the slave;'
assign:task36:'quiz_1;PBN;;A trivia quiz with consequences;'
assign:task37:'running_on_spot_nipples_clamped;PBN;Nippleclamps,Fitness,Pain
-N;Running on the spot with clamped nipples;'
assign:task38:'sissy_lessons_1;PBN;Sissy;Lessons to become a better sissy or
slut;'
assign:task39:'slaves_custom_task;PBN;;The slave defines and later maybe
does a self defined task which is close to their limits;'
assign:task40:'spanking_1;PBN;Crop,Pain-N;Runa has the slave spank a
bodypart of Runas choice;'
assign:task41:'stroke_for_feet;P;;Brainwashing into liking feet by having to
stroke for them'
assign:task42:'stroke_for_marked_feet;P;NotinChastity;Brainwashing into
liking feet by having to look very closely at them'
assign:task43:'stroke_for cocks;PBN;;Brainwashing into liking cocks by
having to stroke for them'
assign:task44:'stroke_for_marked cocks;PBN;;Brainwashing into liking cocks
by having to look very closely at them'
assign:task45:'submission;PBN;;Showing submission on knees and head on the
floor;'
assign:task46:'temp_exp_1;PBN;Online;Chance to get temp exposed;'
assign:task47:'Tethered;PBN;Bondage;Being bound by the balls to something;'
assign:task48:'Thirsty;PBN;Handcuffs,Bondage,Watersport;Drinking pee;'
assign:task49:'video_viewing;PBN;;Watching porn very closely;'
assign:task50:'virtual_pillory_1;PBN;;The slave has to spend time in a
virtual pillory as another slave has defined it;'
assign:task51:'wax_on_penis;P;NotinChastity,Candle,Pain-G;Runa has some fun
with wax on the slaves genitals;'
assign:task52:'waxnipples;PBN;Candle,Pain-N;Runa has some fun with wax on
the slaves nipples;'
assign:task53:'whipping_with_red_lines;PBN;Pain-N;Whipping the butt and
having to generate visible red lines;'
assign:task54:'workout_1;PBN;Fitness;Squats and optional nipple pain;'
assign:task55:'worship cocks;PBN;;Brainwashing into liking cocks by having
to worship them;'
assign:task56:'worship_feet;P;;Brainwashing into liking feet by having to
worship them;'
assign:maxTasks:56
```

```

;
; NOTE: If you want to add more tasks, those tasks need to include the gen
library and call the init and the finish functions as follows:
; assign:taskName:'name_of_the_task'
; use:gen
; call:taskQueue2Init
; ... the actual task ...
; call:taskQueue2Finish
;
;
;=====
;===== defining variables =====
;=====
assign:taskName:'market_1'
use:gen
waitm:300
call:taskQueue2Init
assign:aTaskWasBought:false
assign:taskPriceDecreaseHours:36
assign:maxTaskPrice:10
assign:minTaskPrice:5
call:loadTaskPrice
waitm:300
peerreview:false
assign:curMoodBalance:M00D+5
assign:gFromTaskMoodReduction:0
assign:globalSlaveListVarNeedsUpdate:false
;
;=====
;===== defining functions =====
;=====
;-----
function:loadTaskPrice
;-----
loadgtemp1:MARKET_1_TASK_PRICE
assign:curTaskPrice:TEMP1
loadgtemp1:MARKET_1_LAST_BUY_TIME
assign:lastBuyTime:TEMP1
assign:hoursSinceLastBuy:Math.floor((TIMENOW-$lastBuyTime)/1000/60/60)
if: $hoursSinceLastBuy >= $taskPriceDecreaseHours
    assign:curTaskPrice:$curTaskPrice-
Math.floor($hoursSinceLastBuy/$taskPriceDecreaseHours)
fi:
;
if: $curTaskPrice < $minTaskPrice
    assign:curTaskPrice:$minTaskPrice
fi:
;
if: $curTaskPrice > $maxTaskPrice
    assign:curTaskPrice:$maxTaskPrice

```

```

fi:
  assign:minAllowedMood:20+$curTaskPrice
end:
;
;-----
function:slowDownIntro
;-----
image:taskimg/assistant_at_desk5.jpg
show:One moment, I'll be there for you in a few seconds.
waitm:5000
end:
;
;-----
function:welcomeScreen
;-----
call:slowDownIntro
assign:welcomeText1:'Welcome to the slave task market. '
assign:else1:''
call:getAvailableSlaveNames
label:startOfWelcomeScreenLabel
image:taskimg/assistant_at_desk2.jpg
if: $thisSlaveIsAvailable
  settemp1:'<b>Yes (up to $thisSlaveTasksLeft tasks left)</b>'
else:
  settemp1:'<b>No</b>'
fi:
  show:$welcomeText1What$else1 would you like to do?<p>Available on the
market&colon; TEMP1<br>Current mood balance&colon;
<b>$curMoodBalance</b><br>Price per task&colon; <b>$curTaskPrice</b>
  choice:Buy a task for a slave,Change my own availability,Show my
outstanding tasks,Help,Done
  assign:welcomeScreenChoice:LASTASK
  if: $welcomeScreenChoice == 1
    if: $aTaskWasBought
      show:You cannot buy more than one task at once, but feel free to
return to the market to buy another task.
      button:Ok
    else:
      call:showSlaveSelection
  fi:
fi:
if: $welcomeScreenChoice == 2
  call:changeOwnAvailability
fi:
if: $welcomeScreenChoice == 3
  call:showOutstandingTasks
fi:
if: $welcomeScreenChoice == 4
  call:showHelpScreen
fi:
if: $welcomeScreenChoice != 5

```

```

    assign:welcomeText1:''
    assign:else1:' else'
    goto:startOfWelcomeScreenLabel
  fi:
end:
;
;-----
function:showOutstandingTasks
;-----
loadtemp1:TASKS_TODO_BY_ME
assign:myOpenTasks:'TEMP1'
assign:noOfElements:'$myOpenTasks'.split(';').length-1
if: $noOfElements == 0
  show:You don't have any outstanding tasks.
  button:Thank you
  goto:endOfShowOutstandingTasks
fi:
assign:displayList:''
assign:counter:0
label:startOfShowOutstandingTasks
assign:curElement:'$myOpenTasks'.split(';')[$counter]
assign:curTaskName:'$curElement'.split(',')[0]
assign:curTaskDeadline:'$curElement'.split(',')[1]
settemp1:$curTaskDeadline
assign:fullDaysLeft:Math.floor((TEMP1-TIMENOW)/(24*60*60*1000))
assign:fullHoursLeft:Math.ceil((TEMP1-TIMENOW)/(60*60*1000))
assign:fullHoursLeft:$fullHoursLeft - ($fullDaysLeft * 24)
if: $fullHoursLeft == 24
  assign:fullDaysLeft:$fullDaysLeft+1
  assign:fullHoursLeft:0
fi:
assign:timeLeftString:''
if: $fullDaysLeft == 1
  assign:timeLeftString:'$fullDaysLeft day and '
fi:
if: $fullDaysLeft > 1
  assign:timeLeftString:'$fullDaysLeft days and '
fi:
assign:timeLeftString:'$timeLeftString$fullHoursLeft hours'
assign:displayList:'$displayList<li>$curTaskName ($timeLeftString
left)</li>'
assign:counter:$counter+1
if: $counter < $noOfElements
  goto:startOfShowOutstandingTasks
fi:
show:The following tasks are outstanding for
you&colon;<ul>$displayList</ul>
button:Thank you
label:endOfShowOutstandingTasks
end:
;

```

```
;-----  
function:showHelpScreen  
;-----  
  show:On the <b>Slave Task Market</b> you can buy tasks for fellow slaves.  
I will message you, once I have informed them about the task (usually this  
happens when they run a random task). From that time on, they have 20 days  
to complete the task. I will message you again, once they ran it. If they  
don't complete the task in time, they will lose 5 mood.  
  button:Next  
  show:You can also put yourself up on the market, so others can buy tasks  
for you. When someone selects you, they will see a selection of tasks that  
are within your limits to choose from. So if your limits change, or if you  
enter chastity or denial, it would be smart to visit the market again.  
Everytime you open the market, I will update the tasks available for you  
based on your current limits.  
  button:Next  
  show:The price to buy a task is dynamic. If nobody has bought a task in a  
while, it decreases. If a task got bought, it increases. Right now the price  
is $curTaskPrice mood per task.  
  button:Ok  
end:  
;  
;-----  
function:changeOwnAvailability  
;-----  
; input: $thisSlaveIsAvailable, $thisSlaveLockedSettingsUntil  
; output: $thisSlaveIsAvailable, $thisSlaveLockedSettingsUntil,  
$thisSlaveTasksLeft  
;-----  
if: $thisSlaveIsAvailable && $thisSlaveLockedSettingsUntil != 0  
  assign:timeLeftString:''  
  settemp1:$thisSlaveLockedSettingsUntil  
  assign:fullDaysLeft:Math.floor((TEMP1-TIMENOW)/(24*60*60*1000))  
  assign:fullHoursLeft:Math.ceil((TEMP1-TIMENOW)/(60*60*1000))  
  assign:fullHoursLeft:$fullHoursLeft - ($fullDaysLeft * 24)  
  if: $fullHoursLeft == 24  
    assign:fullDaysLeft:$fullDaysLeft+1  
    assign:fullHoursLeft:0  
  fi:  
  if: $fullDaysLeft == 1  
    assign:timeLeftString:'$fullDaysLeft day and '  
  fi:  
  if: $fullDaysLeft > 1  
    assign:timeLeftString:'$fullDaysLeft days and '  
  fi:  
  assign:timeLeftString:'$timeLeftString$fullHoursLeft hours'  
  image:taskimg/assistant_at_desk4.jpg  
  show:I'm sorry, this setting is still locked for $timeLeftString.  
  button:Gulp  
  goto:endOfChangeOwnAvailabilityLabel  
fi:
```

```

;
;
if: $thisSlaveIsAvailable && $thisSlaveLockedSettingsUntil == 0
  show:<b>You are currently available</b> on the Slave Task Market. What
do you want to do?
  choice:I want to stay available,Please remove me
  if: LASTASK == 2
    assign:thisSlaveIsAvailable:false
    assign:thisSlaveLockedSettingsUntil:0
    assign:thisSlaveTasksLeft:0
    call:updateThisSlaveInGlobalSlaveList
    image:taskimg/assistant_at_desk5.jpg
    show:Ok, I have removed you from the market. Keep in mind that you
still have to do any outstanding tasks, if there are any.
    button:Thank you
  fi:
  goto:endOfChangeOwnAvailabilityLabel
fi:
;
;
if: !$thisSlaveIsAvailable
  show:<b>You are currently not available</b> on the Slave Task Market.
What do you want to do?
  choice:Please make me available,I want to stay unavailable
  if: LASTASK == 1
    assign:thisSlaveIsAvailable:true
    image:taskimg/assistant_at_desk5.jpg
    show:Ok, you are now available. How many tasks can be assigned to you
at most while you are available? Remember, you have 20 days to complete each
task after it is assigned.
    choice:5,10,15,20,25,30,35,40
    assign:thisSlaveTasksLeft:5*LASTASK
    asknumber:For how many days do you want to lock your availability
settings ('0' for no lock)? Note that once all $thisSlaveTasksLeft tasks
have been bought, the lock will also be lifted.
    image:taskimg/assistant_at_desk2.jpg
    if: LASTASK == 0
      show:No problem, I will not lock these settings.
      assign:thisSlaveLockedSettingsUntil:0
    else:
      assign:chosenDays:LASTASK
      if: $chosenDays < 1
        assign:chosenDays:1
      fi:
    assign:thisSlaveLockedSettingsUntil:(TIMENOW+($chosenDays*(24*60*60*1000)))
    show:Ok, I have locked these settings for $chosenDays days.
    fi:
    button:Thank you
    call:updateThisSlaveInGlobalSlaveList
  fi:
fi:

```

```
label:endOfChangeOwnAvailabilityLabel
end:
;
;-----
function:showSlaveSelection
;-----
if: '$availableSlaveNames' == ''
    show:Currently there are no other slaves available on the market.
    button:Ok
    goto:endOfShowSlaveSelectionLabel
fi:
if: $curMoodBalance < $minAllowedMood
    show:A task costs $curTaskPrice mood, but you need to have at least
    $minAllowedMood mood to be able to buy tasks.
    button:Ok, I understand
    goto:endOfShowSlaveSelectionLabel
fi:
label:startOfShowSlaveSelectionLabel
assign:inList:'$availableSlaveNames'
call:getFormattedList
assign:formattedAvailableSlaveNames:'$outList'
asknumber:For which slave should it be? (Enter a number or 0 to
abort)<p>$formattedAvailableSlaveNames
assign:selectedNumber:LASTASK
assign:selectedNumber:Math.floor(Math.abs(selectedNumber))
if: $selectedNumber == 0
    goto:endOfShowSlaveSelectionLabel
fi:
if: $selectedNumber <= '$availableSlaveNames'.split(', ').length
    assign:selectedSlaveName:'$availableSlaveNames'.split(',
')[$selectedNumber-1]
    call:showTaskSelection
else:
    show:<b>$selectedNumber</b> is not in the list.
    button:oh
    goto:startOfShowSlaveSelectionLabel
fi:
label:endOfShowSlaveSelectionLabel
end:
;
;-----
function:showTaskSelection
;-----
call:getAvailableTasksList
label:startOfShowTaskSelectionLabel
assign:inList:'$availableTaskNames'
call:getFormattedList
assign:formattedAvailableTaskNames:'$outList'
asknumber:Which task would you like me to force <b>$selectedSlaveName</b>
to do? (Enter a number or 0 to abort)<p>$formattedAvailableTaskNames
assign:selectedNumber:LASTASK
```

```

    assign:selectedNumber:Math.floor(Math.abs(selectedNumber))
    if: $selectedNumber == 0
      goto:endOfShowTaskSelectionLabel
    fi:
    if: $selectedNumber <= '$availableTaskNames'.split(', ').length
      assign:selectedTaskName:'$availableTaskNames'.split(',
')[$selectedNumber-1]
      call:buyTask
    else:
      show:<b>$selectedNumber</b> is not in the list.
      button:oh
      goto:startOfShowTaskSelectionLabel
    fi:
  label:endOfShowTaskSelectionLabel
end:
;
;-----
function:getFormattedList
;-----
; input: $inList
; output: $outList
;-----
assign:outList:''
assign:noOfElements:'$inList'.split(', ').length
if: $noOfElements == 0
  goto:endOfGetFormattedListLabel
fi:
assign:counter:0
label:startOfGetFormattedListLabel
assign:curElement:'$inList'.split(', ')[$counter]
assign:counter:$counter+1
assign:outList:'$outList<br>$counter. $curElement'
if: $counter < $noOfElements
  goto:startOfGetFormattedListLabel
fi:
label:endOfGetFormattedListLabel
end:
;
;-----
function:getTaskDescription
;-----
; $input: $selectedTaskName
; $output: $taskDescription
;-----
assign:curTaskNo:1
label:startOfGetTaskDescriptionLabel
assign:curTaskId:'task$curTaskNo'
settempl:$curTaskId
assign:curTaskName:'TEMP1'.split(';')[0]
if: '$curTaskName' == '$selectedTaskName'
  assign:taskDescription:'TEMP1'.split(';')[3]

```

```

fi:
assign:curTaskNo:$curTaskNo+1
if:$curTaskNo <= $maxTasks
  goto:startOfGetTaskDescriptionLabel
fi:
if: '$taskDescription' == ''
  assign:taskDescription:'No description available for this task'
fi:
end:
;
;-----
function:buyTask
;-----
  call:getTaskDescription
  show:Do you really want me to force <b>$selectedSlaveName</b> to do the
task <b>$selectedTaskName</b>? This would cost you $curTaskPrice mood (you
currently have $curMoodBalance mood)<p><b>Task description&colon;</b>
$taskDescription
  choice:Yes force them to do it,No I changed my mind
  if: LASTASK == 1
    image:taskimg/assistant_at_desk5.jpg
    show:Alright, <b>$selectedSlaveName</b> will be forced to do the task
<b>$selectedTaskName</b>. I will message you once I have informed them about
the task (usually this happens when they run a random task). From that time
on, they have 20 days to run the task. I will message you again, once they
ran the task.
    assign:gFromTaskMoodReduction:$gFromTaskMoodReduction+$curTaskPrice
    assign:curMoodBalance:$curMoodBalance-$curTaskPrice
    log:Deducting $curTaskPrice mood for buying a task for
$selectedSlaveName
    assign:dTaskName:'$selectedTaskName'
    assign:dTaskToUser:'$selectedSlaveName'
    call:globalScheduleTask
    call:removeOneTaskFromSelectedSlaveInGlobalSlaveList
    broadcast:SNAME just bought a task for another slave. If you also want
to buy a task for someone, just go to the Slave Task Market.
    assign:aTaskWasBought:true
    if: $curTaskPrice < $maxTaskPrice
      assign:curTaskPrice:$curTaskPrice+1
      assign:minAllowedMood:20+$curTaskPrice
    fi:
  else:
    show:No problem, I will not assign the task to them, and you don't have
to pay for it.
  fi:
  button:Thank you
end:
;
;-----
function:getCurSlaveAttributes
;-----

```

```

; input: $curSlaveName, $completeSlaveList
; output: $curSlaveIsAvailable, e.g. true
;         $curSlaveAvailableSince, e.g. 12324365324
;         $curSlaveLockedSettingsUntil, e.g. 12324365324
;         $curSlaveTasksLeft, e.g. 7
;         $curSlaveHas, e.g. 'Penis,Crop,Candle,Pain-N,Pain-G'
;         $curSlaveBody, i.e. 'P' | 'B' | 'N'
;-----
assign:curSlaveIsAvailable:false
assign:curSlaveLockedSettingsUntil:0
assign:curSlaveAvailableSince:0
assign:curSlaveTasksLeft:0
assign:curSlaveHas:''
assign:curSlaveBody:''
assign:noOfExistingSlaves:$completeSlaveList.split(';').length-1
assign:curIndex:-1
label:startOfGetCurSlaveAttributesLabel
assign:curIndex:$curIndex+1
if: $curIndex > $noOfExistingSlaves-1
  goto:endOfGetCurSlaveAttributesLabel
fi:
settemp1:$completeSlaveList.split(';')[$curIndex]
settemp2:TEMP1.split('!')[0]
if: TEMP2 == $curSlaveName
  assign:curSlaveIsAvailable:true
  assign:curSlaveTasksLeft:TEMP1.split('!')[1]
  assign:curSlaveAvailableSince:TEMP1.split('!')[2]
  assign:curSlaveLockedSettingsUntil:TEMP1.split('!')[3]
  assign:curSlaveBody:TEMP1.split('!')[4]
  assign:curSlaveHas:TEMP1.split('!')[5]
  goto:endOfGetCurSlaveAttributesLabel
fi:
goto:startOfGetCurSlaveAttributesLabel
label:endOfGetCurSlaveAttributesLabel
if: $curSlaveLockedSettingsUntil < TIMENOW
  assign:curSlaveLockedSettingsUntil:0
fi:
if: $curSlaveTasksLeft == 0
  assign:curSlaveIsAvailable:false
  assign:curSlaveLockedSettingsUntil:0
  assign:curSlaveHas:''
  assign:curSlaveBody:''
fi:
end:
;
;-----
function:getAvailableSlaveNames
;-----
; input: $completeSlaveList
; output: $availableSlaveNames, e.g. 'Thomas3, JohnF, Lisa'
;         $thisSlaveIsAvailable, e.g. true

```

```

;      $thisSlaveLockedSettingsUntil, e.g. 1751253234534
;      $thisSlaveAvailableSince, e.g. 1751253234534
;      $thisSlaveTasksLeft, e.g. 10
;-----
waitm:300
assign:availableSlaveNames:''
assign:thisSlaveIsAvailable:false
assign:thisSlaveAvailableSince:TIMENOW
assign:thisSlaveLockedSettingsUntil:0
assign:thisSlaveTasksLeft:0
assign:noOfExistingSlaves:'$completeSlaveList'.split(';').length-1
assign:curIndex:-1
label:startOfGetAvailableSlaveNamesLabel
assign:curIndex:$curIndex+1
if: $curIndex > $noOfExistingSlaves-1
  goto:endOfGetAvailableSlaveNamesLabel
fi:
assign:curSlaveName:'$completeSlaveList'.split(';')[$curIndex].split('!')[0]
call:getCurSlaveAttributes
waitm:150
if: '$curSlaveName' == 'SNAME'
  assign:thisSlaveIsAvailable:$curSlaveIsAvailable
  assign:thisSlaveLockedSettingsUntil:$curSlaveLockedSettingsUntil
  assign:thisSlaveAvailableSince:$curSlaveAvailableSince
  assign:thisSlaveTasksLeft:$curSlaveTasksLeft
fi:
if: '$curSlaveName' != 'SNAME' && $curSlaveIsAvailable
  assign:availableSlaveNames:'$availableSlaveNames, $curSlaveName'
fi:
goto:startOfGetAvailableSlaveNamesLabel
label:endOfGetAvailableSlaveNamesLabel
assign:availableSlaveNames:'$availableSlaveNames'.slice(2)
end:
;
;-----
function:getAvailableTasksList
;-----
; input: $curSlaveName
; output: $availableTaskNames, e.g. 'spanking_1, confession_1'
;-----
assign:curSlaveName:'$selectedSlaveName'
call:getCurSlaveAttributes
assign:curTaskNo:1
assign:availableTaskNames:''
label:startOfGetAvailableTasksListLabel
assign:curTaskId:'task$curTaskNo'
settemp1:$curTaskId
assign:curTaskName:'TEMP1'.split(';')[0]
assign:curTaskBody:'TEMP1'.split(';')[1]
assign:curTaskRequires:'TEMP1'.split(';')[2]
assign:requirementsSize:'$curTaskRequires'.split(',').length

```

```

assign:curRequirementIndex:0
label:startOfAnalyzeCurTaskRequirements
settemp1:'$curTaskRequires'.split(',')[$curRequirementIndex]
assign:curRequirementName:'TEMP1'
if:!'$curSlaveHas'.includes('$curRequirementName')
  goto:dontAddTaskLabel
fi:
assign:curRequirementIndex:$curRequirementIndex+1
if: $curRequirementIndex < $requirementsSize
  goto:startOfAnalyzeCurTaskRequirements
fi:
if: !'$curTaskBody'.includes('$curSlaveBody')
  goto:dontAddTaskLabel
fi:
assign:availableTaskNames:'$availableTaskNames, $curTaskName'
label:dontAddTaskLabel
assign:curTaskNo:$curTaskNo+1
if:$curTaskNo <= $maxTasks
  goto:startOfGetAvailableTaslsListLabel
fi:
  ; make the order a bit more random: Keep alphabetical order, but with a
  random start (e.g. X, Y, Z, A, B, C...)
  assign:cutAtPlace:Math.floor(Math.random() * '$availableTaskNames'.length)
  assign:cutAtPlace:'$availableTaskNames'.indexOf(', ', $cutAtPlace)
  if: $cutAtPlace != -1
    assign:firstPart:'$availableTaskNames'.substring(0, $cutAtPlace)
    assign:secondPart:'$availableTaskNames'.substring($cutAtPlace)
    assign:availableTaskNames:'$secondPart$firstPart'
  fi:
  assign:availableTaskNames:'$availableTaskNames'.slice(2)
end:
;
;-----
function:removeOneTaskFromSelectedSlaveInGlobalSlaveList
;-----
  assign:indexOfStart:'$completeSlaveList'.indexOf('$selectedSlaveName!')
  assign:indexOfStart:$indexOfStart+'$selectedSlaveName'.length+1
  assign:indexOfEnd:'$completeSlaveList'.indexOf('!', $indexOfStart)
assign:curTasksLeftOld:'$completeSlaveList'.substring($indexOfStart, $indexOf
End)
  assign:curTasksLeftNew:$curTasksLeftOld-1
assign:completeSlaveList:'$completeSlaveList'.replace('$selectedSlaveName!$c
urTasksLeftOld', '$selectedSlaveName!$curTasksLeftNew')
  assign:globalSlaveListVarNeedsUpdate:true
end:
;
;-----
function:updateThisSlaveInGlobalSlaveList
;-----
  assign:indexOfStart:'$completeSlaveList'.indexOf('SNAME!')
  assign:oldThisSlaveHas:''

```

```

assign:actuallyUpdatedThisSlaveHas:false
if: $indexOfStart != -1
  assign:indexOfEnd:'$completeSlaveList'.indexOf(';',$indexOfStart)
  assign:substringBefore:'$completeSlaveList'.substring(0,$indexOfStart)
assign:substringAfter:'$completeSlaveList'.substring($indexOfEnd+1,9999999)
assign:oldThisSlaveHas:'$completeSlaveList'.substring($indexOfStart,$indexOfEnd)
  assign:completeSlaveList:'$substringBefore$substringAfter'
  assign:curIndex:'$oldThisSlaveHas'.split('!').length
  assign:curIndex:$curIndex-1
  assign:oldThisSlaveHas:'$oldThisSlaveHas'.split('!')[$curIndex]
fi:
if: $thisSlaveIsAvailable
  assign:thisSlaveAvailableSince:TIMENOW
  call:readThisSlaveHas
assign:completeSlaveList:'SNAME!$thisSlaveTasksLeft!$thisSlaveAvailableSince!$thisSlaveLockedSettingsUntil!$thisSlaveBody!$thisSlaveHas;$completeSlaveList'
  assign:actuallyUpdatedThisSlaveHas: '$oldThisSlaveHas' != '$thisSlaveHas'
  fi:
  assign:globalSlaveListVarNeedsUpdate:true
end:
;
;-----
function:loadCompleteSlaveList
  ;-----
  ; output: $completeSlaveList, e.g.
  'Adam!4!1234!5678!P!Crop;TittyTwister!13!7654!2344!P!Crop,Candle,Pain-N,Pain-G;Thomas3!12!2345!7894!P!Crop,Pain-N,Pain-G;JohnF!4!1234!456!P!Crop,Pain-N,Pain-G,Nippleclamps;Lisa!99999!34544!68745!B!Crop,Candle,Pain-N,Pain-G;'
  ; format:
  'SNAME!<$curSlaveTasksLeft>!<$curSlaveAvailableSince>!<$curSlaveLockedSettingsUntil>!<$curSlaveBody>!<$curSlaveHas>;...'
  ;-----
  loadgtemp1:MARKET_1_COMPLETE_SLAVE_LIST
  assign:completeSlaveList:'TEMP1'
end:
;
;-----
function:maybeSaveCompleteSlaveList
  ;-----
  ; Also update the slave list if the user's limits have changed
if: !$globalSlaveListVarNeedsUpdate
  assign:oldCompleteSlaveList:'$completeSlaveList'
  call:updateThisSlaveInGlobalSlaveList
  if: $actuallyUpdatedThisSlaveHas
    assign:globalSlaveListVarNeedsUpdate:true
  else:
    assign:globalSlaveListVarNeedsUpdate:false

```

```

    fi:
fi:
if: $globalSlaveListVarNeedsUpdate
  show:Just a second, I need to enter some entries in the system.
  image:taskimg/assistant_at_desk5.jpg
  waitm:2000
  settemp1:'$completeSlaveList'
  savegtemp1:MARKET_1_COMPLETE_SLAVE_LIST
  show:All done, see you next time
  image:taskimg/assistant_at_desk2.jpg
  wait:1
fi:
end:
;
;-----
function:readThisSlaveHas
;-----
; output: $thisSlaveHas, e.g. 'NotinChastity,Candle'
;       $thisSlaveBody, i.e. 'P' | 'B' | 'N'
;-----
assign:thisSlaveBody:'N'
; ----- Body: -----
if: HAS_PENIS
  assign:thisSlaveBody:'P'
fi:
if: HAS_BREASTS
  assign:thisSlaveBody:'B'
fi:
;
assign:thisSlaveHas:''
; ----- Chastity/Denial: -----
if: INCHASTITY
  assign:thisSlaveHas:'$thisSlaveHas,InChastity'
fi:
if: !INCHASTITY
  assign:thisSlaveHas:'$thisSlaveHas,NotinChastity'
fi:
if: !DENIED
  assign:thisSlaveHas:'$thisSlaveHas,CanCum'
fi:
; ----- Toys: -----
;TODO: Remoteestim
;TODO: Remoteshock
;TODO: Remotelock
if: T_BUTTPLUG
  assign:thisSlaveHas:'$thisSlaveHas,Buttplug'
fi:
if: T_CANDLE
  assign:thisSlaveHas:'$thisSlaveHas,Candle'
fi:
if: T_CROP

```

```
    assign:thisSlaveHas:'$thisSlaveHas,Crop'  
fi:  
if: T_DILDO  
    assign:thisSlaveHas:'$thisSlaveHas,Dildo'  
fi:  
if: T_ESTIM  
    assign:thisSlaveHas:'$thisSlaveHas,E-Stim'  
fi:  
if: T_GAG  
    assign:thisSlaveHas:'$thisSlaveHas,Gag'  
fi:  
if: T_HANDCUFFS  
    assign:thisSlaveHas:'$thisSlaveHas,Handcuffs'  
fi:  
if: T_HUMBLER  
    assign:thisSlaveHas:'$thisSlaveHas,Humbler'  
fi:  
if: T_NIPPLECLAMPS  
    assign:thisSlaveHas:'$thisSlaveHas,Nippleclamps'  
fi:  
; -----Limits: -----  
if: L_ANAL > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Anal'  
fi:  
if: L_ATM > 1  
    assign:thisSlaveHas:'$thisSlaveHas,ATM'  
fi:  
if: L_BONDAGE > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Bondage'  
fi:  
if: L_CHASTITY > 1  
    assign:thisSlaveHas:'$thisSlaveHas,ChastityDevice'  
fi:  
if: L_CUM > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Cum'  
fi:  
if: L_FACE > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Face'  
fi:  
if: L_FITNESS > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Fitness'  
fi:  
if: L_FINDOM > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Financial'  
fi:  
if: L_ONLINE > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Online'  
fi:  
if: L_OUTDOOR > 1  
    assign:thisSlaveHas:'$thisSlaveHas,Outdoor'  
fi:
```

```

if: L_EXHIB > 1
  assign:thisSlaveHas:'$thisSlaveHas,Exhibitionism'
fi:
if: L_DENIAL > 1
  assign:thisSlaveHas:'$thisSlaveHas,Orgasm'
fi:
if: L_PAIN_G > 1
  assign:thisSlaveHas:'$thisSlaveHas,Pain-G'
fi:
if: L_PAIN_N > 1
  assign:thisSlaveHas:'$thisSlaveHas,Pain-N'
fi:
if: L_INFO > 1
  assign:thisSlaveHas:'$thisSlaveHas,PInfo'
fi:
if: L_SISSY > 1
  assign:thisSlaveHas:'$thisSlaveHas,Sissy'
fi:
if: L_SHAVE > 1
  assign:thisSlaveHas:'$thisSlaveHas,Shaving'
fi:
if: L_WATERSPORT > 1
  assign:thisSlaveHas:'$thisSlaveHas,Watersport'
fi:
assign:thisSlaveHas:'$thisSlaveHas'.slice(1)
end:
;
;-----
function:maybeSaveTaskPrice
;-----
if: $aTaskWasBought
  settemp1:$curTaskPrice
  savegtemp1:MARKET_1_TASK_PRICE
  settemp1:TIMENOW
  savegtemp1:MARKET_1_LAST_BUY_TIME
fi:
end:
;
;=====
;===== start of the task =====
;=====
call:loadCompleteSlaveList
call:welcomeScreen
call:maybeSaveCompleteSlaveList
call:maybeSaveTaskPrice
call:taskQueue2Finish
;

```

</code>

From:

<https://play-link.com/wiki/> - **PlayLink**

Permanent link:

https://play-link.com/wiki/doku.php?id=tasks:slave_task_market



Last update: **2026/02/11 21:16**