

```
Title:thomas_test
Sex: Female T Male T Trans T
Require:
Limits:
use:chastity
use:gen
;
label:StartTask
show:Select an option
choice:test code, delete review array elements, clear chastity, show review
queue, show Task Log, delete Task log, show to-do list, next
assign:TestChoice:LASTASK
if:(TestChoice == 1)
    button:testing code
    ; ***** test code start
    show:Ok, when you run the cumming task, you are allowed to unlock then,
but you have to re-lock again immediately after cumming. I have added 8
hours of chastity time, during which you should run the controlled cumming
task. See you back here in a while
    button:Yes DTITLE
    show:Oh, and to make one thing clear. You MUST lock up after you have done
the controlled cumming, even if you think its only for an hour. Yes it might
waste a tag, but also there might always be a surprise waiting for you. So
you are NOT allowed to just stay unlocked because it was your last
controlled cum, or you think it was
    button:I will lock up again DTITLE
    log:Slave has been assigned controlled cumming tasks. He is not allowed to
unlock. After running the controlled cumming task, he MUST lock up again, no
matter what. Please also check for that.
    settemp1:1
    ; schedule validation tasks in case slave is locked up for longer e.g. by
controlled cumming task
    savetemp1:controlledCummingUnlock
    ; ***** test code end
    button:done testing code
fi:
if:(TestChoice == 2)
    show:WARNING. You will delete some entries in the club review queue. Do
you want this
    button:yes
    loadglist:chastityClubReviews
    asknumber:Entry number (starting count with 1)
    settemp3:(LASTASK - 1)
    listdelete:TEMP3
    saveglist:chastityClubReviews
    ; ****\*\*\*\*\* Update number of pending reviews
    assign:numberOfReviews:(List.length)
    if:($numberOfReviews == 0)
        goto:exitTask
    fi:
    settemp1:0
```

```
loop:$numberOfReviews
  settemp2:List[(LOOPVAR - 1)]
  split:'TEMP2'
  if:(SPLIT5 == false)
    settemp1:TEMP1 + 1
  fi:
endloop:
savegtemp1:chastityClubNewVideos
fi:
if:(TestChoice == 3)
  settemp1:''
  savetemp1:chastityTagNumber
  savetemp1:chastityLock
  savetemp1:chastityStartTime
  savetemp1:chastityStartTimeP
  savetemp1:unlock_pending
  savetemp1:pending-unlock-time
  savetemp1:chastityLockPic
  savetemp1:chastityLastValidation
  savetemp1:chastityMustCumFast
fi:
if:(TestChoice == 4)
  loadglist:chastityClubReviews
  assign:numberOfReviews:(List.length)
  if:(($numberOfReviews == 0))
    show:There are no reviews to see
    button:ok
    goto:exitTask
  fi:
  loop:$numberOfReviews
    settemp2:List[(LOOPVAR - 1)]
    split:'TEMP2'
    show:Entry LOOPVAR TEMP2
    button:ok
  endloop:
  loadgtemp1:chastityClubNewVideos
  show:Number of pending reviews:TEMP1
  button:ok
fi:
if:(TestChoice == 5)
  show:Filter?
  choice:No, CC-Members, Other
  loadglist:logTaskList
  assign:numberOfLogs:(List.length)
  if:(($numberOfLogs == 0))
    show:There are no logs to see
    button:ok
    goto:exitTask
  fi:
  loop:$numberOfLogs
    settemp2:List[(LOOPVAR - 1)]
```

```

    split:'TEMP2'
    if:('TEMP2' != '')
        ; && (SPLIT2 > (STIME - 432000))
        if:(LASTASK == 1)
            call:showLogEntry
        fi:
        if:((LASTASK == 2) && (('SPLIT3' == 'Thomas3') || ('SPLIT3' ==
'Chissoku') || ('SPLIT3' == 'JohnF') || ('SPLIT3' == 'lisa') || ('SPLIT3' ==
'Loser1') || ('SPLIT3' == 'TittyTwister'))))
            call:showLogEntry
        fi:
        if:((LASTASK == 3) && !SPLIT5)
            call:showLogEntry
        fi:
        fi:
        iframe:CLEAR
    endloop:
fi:
if:(TestChoice == 6)
    show:WARNING. You will delete all task log
    button:yes
    loadglist:logTaskList
    setlist:'EMPTY,STIME,SNAME,TID,HAS_PENIS'
    saveglist:logTaskList
    ; loadlist:toDoTasks
    ; setlist:'EMPTY,0,0,90000000000'
    ; savelist:toDoTasks
fi:
if:(TestChoice == 7)
    loadlist:toDoTasks
    assign:numberOfTasks:(List.length)
    loop:$numberOfTasks
        settemp1:List[(LOOPVAR - 1)]
        split:'TEMP1'
        show:SPLIT1 SPLIT2 SPLIT3 SPLIT4
        button:next
    endloop:
fi:
if:(TestChoice != 8)
    goto:StartTask
fi:
choice:run task, delete to-do list, exit, back
assign:TestChoice:LASTASK
if:(TestChoice == 1)
    show:about to schedule a task
    button:ok
    dmsg:0:Jennifer:Time for your task
    dtask:0:Jennifer:chastity_edging:1
fi:
if:(TestChoice == 2)
    show:removing to-do list items

```

```

loadlist:toDoTasks
settemp1:'Empty, 100, 100, 5000000000'
setlist:'TEMP1'
savelist:toDoTasks
button:ok
fi:
if:(TestChoice != 3)
    goto:StartTask
fi:
;
label:exitTask
;
;-----
-----
function:scheduleToDoTask
    ;** Schedules a task as standard JIC delayed task AND stores the need to
run the task in a list
    ;** accessible by tasks. This function will become obsolete once the
standard JIC delayed task
    ;** system will be able to punish slaves for ignoring to-do tasks
    ;** Expected arguments:$taskName, $taskDelay, $taskGrace
    ;** Return values: none
    ;** Modifies TEMP1 variable.
dtask:$taskDelay:Jennifer:$taskName:$taskGrace
loadlist:toDoTasks
settemp1:'$taskName, $taskDelay, $taskGrace, STIME'
listadd:'TEMP1'
savelist:toDoTasks
end:
;-----
-----
function:validateToDoTasks
    ;** Validates whether any to-do tasks of the slave are overdue.
    ;** Assigns a punishment in case one or more are overdue
    ;** Expected arguments: none. Returned value: none
    ;** Modifies TEMP1 variable.
loadlist:toDoTasks
assign:numberOfTasks:(List.length)
assign:punishmentDue:false
loop:$numberOfTasks
    settemp1:List[(LOOPVAR - 1)]
    split:'TEMP1'
    if:(SPLIT4 + ((SPLIT2 + SPLIT3) _ 3600) < STIME)
        show:You are lated doing SPLIT1 task. A punishment will be assigned
        log:Slave is late doing SPLIT1 task. A punishment will be assigned
        button:Yes DTITLE
        assign:punishmentDue:true
    fi:
endloop:
if:punishmentDue
    show:Punishment logic will be inserted here

```

```

    button:ok
    fi:
end:
;-----
-----
function:t_logTaskRun
    /** Maintains a log of how often a task has been run.
    /** Removes the task from the task toDo list in case it is on there
    /** Modifies TEMP1 variables
    /** Expected arguments: logTaskName. Returned values: none
    ;
    ; log task run
    loadglist:logTaskList
    settemp1:'$logTaskName,STIME,SNAME,TID,HAS_PENIS'
    listadd:'TEMP1'
    saveglist:logTaskList
    ;
    ; remove task from to-do list in case its on there
    loadlist:toDoTasks
    assign:numberOfTasks:(List.length)
    assign:taskPosition:0
    assign:taskTime:0
    loop:$numberOfTasks
        settemp1:List[(LOOPVAR - 1)]
        split:'TEMP1'
        if(('SPLIT1' == '$logTaskName') && (($taskTime == 0) || (SPLIT4 <
$taskTime)))
            assign:taskPosition: (LOOPVAR - 1)
            assign:taskTime:SPLIT4
        fi:
    endloop:
    if: ($taskTime != 0)
        listdelete:$taskPosition
        savelist:toDoTasks
    fi:
end:
;
function:showLogEntry
    show:Entry LOOPVAR<br>Task SPLIT1<br>Time SPLIT2<br>Slave SPLIT3<br>SPLIT4
SPLIT5
    iframe:https://play-link.com/jennifer/rtask.php?task=SPLIT4
    button:ok
end:
;
function:reviewCCslavesInstructions
    if:'SNAME' == 'Thomas3'
        image:taskimg/chastity/Thomas3 Instructions.jpg
        show:Please verify that the slave has done the following in the
video<br><br><b>1</b> Did he <b>strongly</b> pull on the tag<br><b>2</b> Is
the wire threaded through <b>tube AND piercing</b><br><b>3</b> Did he show
the cut tag <b>with all wires visible</b><br><b>4</b> Did he use his long

```

```

tube</b>
    button:Yes I have verified this
    fi:
end:
;
;-----
-----
function:calculateDayOfMonth
    ; assigns dayOfMonth variable. For example if called on January 10th, it
    will return 10
    ; start with January 1 2025, 00 00 01 GMT
    assign:epochReference:1735689601
    assign:secondsToDays:(24 _ 60 _ 60)
    loop:100
        if: (STIME < epochReference + (31 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1
            goto: exitCalculateDayOfMonth
        else:
            assign:epochReference:epochReference + (31 _ secondsToDays)
        fi:
        if: (STIME < epochReference + (28 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1
            goto: exitCalculateDayOfMonth
        else:
            assign:epochReference:epochReference + (28 _ secondsToDays)
        fi:
        if: (STIME < epochReference + (31 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1
            goto: exitCalculateDayOfMonth
        else:
            assign:epochReference:epochReference + (31 _ secondsToDays)
        fi:
        if: (STIME < epochReference + (30 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1
            goto: exitCalculateDayOfMonth
        else:
            assign:epochReference:epochReference + (30 _ secondsToDays)
        fi:
        if: (STIME < epochReference + (31 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1
            goto: exitCalculateDayOfMonth
        else:
            assign:epochReference:epochReference + (31 _ secondsToDays)
        fi:
        if: (STIME < epochReference + (30 _ secondsToDays))
            assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +

```

```

1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (30 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (31 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (31 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (31 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (31 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (30 _ secondsToDays)) ;sept
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (30 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (31 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (31 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (30 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (30 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (31 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (30 _ secondsToDays)
    fi:
    if: (STIME < epochReference + (31 _ secondsToDays))
        assign:dayOfMonth:Math.trunc((STIME-epochReference) / secondsToDays) +
1      goto: exitCalculateDayOfMonth
    else:
        assign:epochReference:epochReference + (31 \* secondsToDays)
    fi:
endloop:
label:exitCalculateDayOfMonth
end:

```

;

From:

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

Permanent link:

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



Last update: **2025/12/25 06:52**