

```
SET NOCOUNT ON
USE ceiton_flow
SET xAct_abort ON
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

IF (exists (select * from sysobjects where name = N'sp_Dispo.calcJobsTagListe' and type = N'
'P '))
    DROP PROCEDURE [dbo].[sp_Dispo.calcJobsTagListe]
GO

create procedure [dbo].[sp_Dispo.calcJobsTagListe]
    @von datetime
    , @bis datetime
    , @feiertage_string nvarchar(1000) = N''
    , @getPlanZeit bit = 0
    , @isIstIsSollRueckgabe bit = 0
    , @showJobsWithoutWT bit = 0
    , @changeJobServiceTypeTo nchar(1) = N'a'

WITH ENCRYPTION
as

if @von is null or @bis is null
begin

    declare @curr_von1 datetime
    declare @curr_bis1 datetime
    declare @curr_von2 datetime
    declare @curr_bis2 datetime

    select @curr_von1 = min(w.aDate)
        , @curr_bis1 = max(w.aDate)
    from #glbTT017_ValidFrei f
        inner join dispo_services s on f.frei_id = s.service_id
        inner join dispo_services_workdays w on s.service_id = w.service_id
    where w.notWorked = 0
        and s.Deleted = 0

    select @curr_von2 = min(a.von)
        , @curr_bis2 = max(a.bis)
    from #glbTT024_ValidAuftraege t
        inner join dispo_auftraege a on a.auftrag_id = t.auftrag_id

    if @von is null
    begin
        if @curr_von1 is null
            set @curr_von1 = convert(datetime, N'2999-12-31', 120)
        if @curr_von2 is null
            set @curr_von2 = convert(datetime, N'2999-12-31', 120)

        if @curr_von1 < @curr_von2
            set @von = @curr_von1
        else
            set @von = @curr_von2
    end

    if @bis is null
    begin
        if @curr_bis1 is null
            set @curr_bis1 = convert(datetime, N'1900-01-01', 120)
        if @curr_bis2 is null
            set @curr_bis2 = convert(datetime, N'1900-01-01', 120)

        if @curr_bis1 > @curr_bis2
            set @bis = @curr_bis1
        else
            set @bis = @curr_bis2
    end
end
```

end

```
create table #calc_jobsProTag (
  sort_id          int,
  tag              datetime,
  job_typ         nchar(1),
  job_id          int,
  user_id         int,
  otherUser_id    int,
  ist_von         int,
  ist_bis         int,
  ist_pause       int,
  ist_anrechenbar int,
  soll_von       int,
  soll_bis       int,
  soll_pause     int,
  verbindlich    bit,
  isIst          bit default 0,
  isSoll         bit default 0,
  geplant_verbindlich bit,
)
```

```
create table #user_feiertage(
  datum  datetime
)
```

```
create table #all_user(
  user_id          int
, holidayCalendar_id int
, primary key (user_id)
)
```

```
declare @currCalendar    int
declare @systemCalendar  int

set @currCalendar = -1
set @systemCalendar = -2
set @systemCalendar = (Select wert_string
                       From all_internalConfigs
                       Where globalConfigs_id = 1)
```

```
insert into #all_user (user_id, holidayCalendar_id)
select distinct t.user_id, isNull(u.holidayCalendar_id, @systemCalendar)
from #glbTT017_ValidFrei t
     left join all_users u on t.user_id = u.all_userid
```

```
insert into #all_user (user_id, holidayCalendar_id)
select distinct isNull(t.user_id,-1), isNull(u.holidayCalendar_id, @systemCalendar)
from #glbTT024_ValidAuftraege t
     left join all_users u on t.user_id = u.all_userid
     left join #all_user tu on isNull(t.user_id,-1) = tu.user_id
where tu.user_id is null
```

```
declare myCursor cursor local fast_forward
for
select distinct holidayCalendar_id
from #all_user
```

Open myCursor

```
fetch next from myCursor into @currCalendar
while @@fetch_status = 0
begin
```

```

delete from #user_feiertage

insert into #user_feiertage (datum)
Select vhc.datum
From vi_All_holiday4AllCalendar vhc
where vhc.internal_nr = @currCalendar
and vhc.datum between @von and @bis

declare @curr_day          datetime

set @curr_day = [dbo].[fn_All.cutTimeFromDate](@von)

while (datediff(dd, @curr_day, @bis) >= 0)
begin

    insert into #calc_jobsProTag (sort_id, tag, job_typ, job_id, user_id, verbindlich,
geplant_verbindlich, isIst, isSoll)
    select distinct 0,
                    @curr_day,
                    N'a',
                    v.auftrag_id,
                    v.user_id,
                    a.verbindlich,
                    a.geplant_verbindlich,
                    CASE
                        WHEN (a.von <= @curr_day) AND (@curr_day <= a.bis) THEN 1
                        ELSE 0
                    END AS isIst,
                    CASE
                        WHEN (a.geplant_von <= @curr_day) AND (@curr_day <= a.geplant_bis)
THEN 1
                        ELSE 0
                    END AS isSoll
    from #glbTT024_ValidAuftraege v
        inner join #all_user u on u.user_id = isNull(v.user_id,-1)
        left join dispo_auftraege a on v.auftrag_id = a.auftrag_id

        left join #user_feiertage uf on uf.datum = @curr_day
    where ( (@curr_day BETWEEN a.von AND a.bis)
           or ((@curr_day BETWEEN a.geplant_von AND a.geplant_bis) and @getPlanZeit = 1 ))
           and u.holidayCalendar_id = @currCalendar

           and ( ( [dbo].[fn_Help.dateIsWeekend](@curr_day) = 0
                 and uf.datum is null)
               or (a.wochenendarbeit = 1)
               or ( a.geplant_wochenendarbeit = 1
                   and @getPlanZeit = 1)
               )

    set @curr_day = dateadd(d, 1, @curr_day)
end

insert into #calc_jobsProTag (sort_id, tag, job_typ, job_id, user_id, ist_von, ist_bis,
ist_pause, ist_anrechenbar,
                           verbindlich, geplant_verbindlich)
select 0, wd.aDate, 'f', v.frei_id, v.user_id,
       wd.actual_from_min, wd.actual_from_min + wd.actual_duration_min, wd.
actual_break_min, wd.actual_worktime,
       s.actual_isBindingly, s.planned_isBindingly
from #glbTT017_ValidFrei v
    inner join dbo.dispo_services s on v.frei_id = s.service_id
    inner join dispo_services_workdays wd on s.service_id = wd.service_id

```

```
inner join #all_user u on u.user_id = v.user_id
                        and u.holidayCalendar_id = @currCalendar
left join #user_feiertage uf on uf.datum = wd.aDate
where 1=1
and wd.aDate BETWEEN @von and @bis
and s.deleted = 0
and wd.notWorked = 0

and ( ( [dbo].[fn_Help.dateIsWeekend](wd.aDate) = 0
        and uf.datum is null)
      or (s.weekend = 1)
      )

fetch next from myCursor into @currCalendar
end

close myCursor
deallocate myCursor

update #calc_jobsProTag
set ist_von = a.von_min,
    ist_bis = a.von_min + a.dauer_min,
    ist_pause = a.pause_min
from #calc_jobsProTag j
     inner join dispo_auftraege a on j.job_id = a.auftrag_id
                        and (j.job_typ = N'a')

where j.tag between a.von and a.bis
and j.verbindlich = 1

update #calc_jobsProTag
set ist_von = t.geplant_von_min,
    ist_bis = t.geplant_von_min + t.geplant_dauer_min,
    ist_pause = t.geplant_pause_min
from #calc_jobsProTag j
     inner join dispo_AuftTagPlanung t on j.job_id = t.auftrag_id
                        and (j.job_typ = N'a')
                        and j.tag = t.datum

where j.verbindlich = 1

if @getPlanZeit = 1
begin

update #calc_jobsProTag
set soll_von = a.geplant_von_min,
    soll_bis = a.geplant_von_min + a.geplant_dauer_min,
    soll_pause = a.geplant_pause_min
from #calc_jobsProTag j
     inner join dispo_auftraege a on j.job_id = a.auftrag_id
                        and (j.job_typ = N'a')

where j.tag between a.geplant_von and a.geplant_bis
and j.geplant_verbindlich = 1

update #calc_jobsProTag
set soll_von = t.geplant_von_min,
    soll_bis = t.geplant_von_min + t.geplant_dauer_min,
    soll_pause = t.geplant_pause_min
from #calc_jobsProTag j
     inner join dispo_AuftTagPlan_geplant t on j.job_id = t.auftrag_id
                        and (j.job_typ = N'a')
                        and j.tag = t.datum

where j.geplant_verbindlich = 1
end

if @showJobsWithoutWT = 0
begin
```

```
if @getPlanZeit = 0
begin
delete from #calc_jobsProTag
where verbindlich = 1
and (ist_von is null or ist_bis is null)
and not job_typ = N'f'
end
else
begin
delete from #calc_jobsProTag
where (verbindlich = 1 and (ist_von is null or ist_bis is null)
and (geplant_verbindlich = 1 and (soll_von is null or soll_bis is null))
and not job_typ = 'f'
end
end
```

```
UPDATE #calc_jobsProTag
SET job_typ = @changeJobServiceTypeTo
FROM #calc_jobsProTag jt
INNER JOIN dispo_auftraege da on jt.job_id = da.auftrag_id
WHERE jt.job_typ = 'a'
AND da.isService = 1
```

```
if @getPlanZeit = 0
begin

delete from #calc_jobsProTag

where verbindlich = 0
and not job_typ = N'f'
and isSoll = 1 and isIst = 0
end
```

```
else
begin
```

```
delete from #calc_jobsProTag

where verbindlich = 0
and geplant_verbindlich = 0

and not job_typ = N'f'

and isSoll = 0 and isIst = 0
end
```

```
update #calc_jobsProTag
set ist_bis = ist_bis - 1440
where ist_bis > 1440
update #calc_jobsProTag
set ist_von = ist_von - 1440
where ist_von > 1440
update #calc_jobsProTag
set soll_von = soll_von - 1440
where soll_von >= 1440
update #calc_jobsProTag
set soll_bis = soll_bis - 1440
where soll_bis > 1440
```

```
if @isIstIsSollRueckgabe = 0
begin
```

```
if @getPlanZeit = 1
begin
select sort_id, tag, job_typ, job_id, user_id, otherUser_id,
ist_von, ist_bis, ist_pause, ist_anrechenbar, verbindlich,
soll_von, soll_bis, soll_pause, geplant_verbindlich
from #calc_jobsProTag
end
else
begin
select sort_id, tag, job_typ, job_id, user_id, otherUser_id,
```

```
        ist_von, ist_bis, ist_pause, ist_anrechenbar, verbindlich, geplant_verbindlich
    from #calc_jobsProTag
end
end
else
begin

    if @getPlanZeit = 1
    begin
        select sort_id, tag, job_typ, job_id, user_id, otherUser_id,
            ist_von, ist_bis, ist_pause, ist_anrechenbar, verbindlich,
            soll_von, soll_bis, soll_pause, isIst, isSoll, geplant_verbindlich
        from #calc_jobsProTag
    end
    else
    begin
        select sort_id, tag, job_typ, job_id, user_id, otherUser_id,
            ist_von, ist_bis, ist_pause, ist_anrechenbar, verbindlich, isIst, isSoll,
        geplant_verbindlich
        from #calc_jobsProTag
    end
end

drop table #calc_jobsProTag

drop table #user_feiertage
drop table #all_user
```