JOBS

DBMS\_JOBS SYS.DBMS\_SCHEDULER.CREATE\_JOB

(

job\_name => 'UNILAB.UNI\_J\_SAMPLEPLANNER'

,start\_date => TO\_TIMESTAMP\_TZ('2021/03/29 00:00:00.000000 +01:00','yyyy/mm/dd hh24:mi:ss.ff tzh:tzm')

,repeat\_interval => 'FREQ = WEEKLY; INTERVAL = 1'

,end\_date => NULL

,job\_class => 'UNI\_JC\_OTHER\_JOBS'

,job\_type => 'PLSQL\_BLOCK'

,job\_action => 'UNAPIPLAN.SamplePlanner(TRUNC(CURRENT\_TIMESTAMP,''DD''),TRUNC(CURRENT\_TIMESTAMP,''DD'') + 7 - 1/86400,24);'

,comments => NULL

);

cxapp.startalldbjobs

unapiplan.set\_connection

unapigen.setconnection  
client\_id := 'SamplePlanner’  
\_us :=l\_dba\_name = 'UNILAB'; *--&Username;*

a\_password = ‘ ‘

al\_applic := 'acplan’

l\_numeric\_characters := 'DB';

l\_date\_format := ''DDfx/fxMM/RR HH24fx:fxMI:SS';  
l\_time\_zone = SERVER  
l\_up =   
l\_user\_profile =   
l\_language =   
l\_tk =

UNAPIGEN.SETCONNECTION4INSTALL

UNAPIGEN.SETCONNECTION4INSTALL (incl. timezone)

ALGEHELE CONNECTIE-PROCEDURE NA INLOGGEN OP DATABASE.

UNFREQ.StPlanOnlyOnWorkDay

ALS St\_freq\_TP= C

UTST:  
**# FREQ-TP FREQ-UNIT**2 C StPlanOnlyOnWorkDay

1 T HH

240 T WW

36 T YY

1 S Request

6 T MI

LOOP UTST .ST waarbij freq\_tp not in (A.N)

cxapp.stopalldbjobs

DBMS\_SCHEDULER.DROP\_JOB(l\_job\_id)

UTST:   
29 A

283 T

2 C

105067 N

1 S

UNAPIPLAN.StartSamplePlanner

**UNAPIPLAN.SamplePlanner**

UTSYSTEM  
DBA\_NAME = UNILAB

JOBS\_DATE\_FORMAT = DDfx/fxMM/RR HH24fx:fxMI:SS

UTSYSTEM  
**SCPLAN\_SCCREATEPG = ON SAMPLE CREATION**

**SCPLAN\_SCCREATEIC =  ON SAMPLE CREATION**

Fetch default creation-flags

ANDERS St\_freq\_**TP= T/S**

UNAPIAUT.EvalAssignmentFreq(st)

ALS PLAN-NEW

UNAPISC.PlanSample

END LOOP UTST .ST

LOOP rows

END LOOP rows

UNFREQ.OnlyOnWorkDay

MO/DI/WO/DO/VR

Freq-unit: MM/HH/DD/WW/MM/MF/YY/YF

UNAPISC.CreateSample

UNAPIGEN.GetMaxSamples

UNAPIAUT.GETSCAUTHORISATION

Haal DATA op

UTST

UTRQ

UTSD

Insert sc

UTSC

UNAPIEV.INSERTEVENT('PlanSample', sc') (event-tp: **'SamplePlanned'**)

UTSCHS

RQ/SC gevuld: UNAPIRQ2.INITANDSAVERQSCATTIBUTES

SD/PT gevuld: UNAPISD2.INITANDSAVESDCELLSCATTIBUTES

Inherit-au gevuld: INSERT into UTSCAU

LOOP STGK-rows (utstgk + utgksc)

END LOOP STGK-rows

insert STGK

UTSCGK

UNAPIIC.CREATESCINFODETAILS

ALS L\_CREATE\_IC = “ON SAMPLE CREATION”

UNAPISC.CREATESCANALYSEDETAILS

UNAPIAUT.CALCULATEDELAY

ALS L\_DELAY gevuld

UNAPIEV.INSERTTIMEDEVENT(event-type=ScActivate)

UTDELAY(sc)

LOOP deadlock-detectie (3sec)

END LOOP deadlock-detectie (min 3sec)

WHILE curr-date < end-date

END WHILE curr-date < end-date

ST-verwerkt, werkt UTST bij (last-sched-date)

Sleep(10)

UTST

Nog onduidelijk met welke waardes worden gebruikt voor SAMPLE-PLANNER.

Welke SC wordt hier geinsert? En wat wordt er dan via EVENT-manager aangemaakt?

SELECT \* FROM **utst**

WHERE freq\_tp NOT IN ('A', 'N')

AND VERSION\_IS\_CURRENT = 1;

UTEV

**UTEV: UPDATE UTEV SET D.EVMGR\_NAME = 'U4EVMGR'**

UNAPIEV2.EVENTMANAGER(L\_TR\_SEQ)

UNAPIEV.EXECUTEACTION(UNAPIEV.P\_AF)

UNACTION.L\_AF

UNAPIEV.L\_OBJ\_CURSOR := DBMS\_SQL.OPEN\_CURSOR;

Loop UTEV-TX-EVENTS

UNAPIEV.EVALUATEEVENTDETAILS

RNDAPIEV.HandleRnDEvents;

INDIEN POSTPONED?

INDIEN not POSTPONED?

UNAPIEV.EVALLIFECYCLE

UNAPIEV.UPDATEOPALOBJECTRECORD(UNAPIEV.P\_SS\_TO)

UNACCESS.INITOBJECTACCESSRIGHTS

UPDATE UTRQ

UNAPIEV.EXECUTETRACTIONS

UNACTION.<AF>

UNAPIEV.P\_EV\_REC

UTLCTR

TR\_SEQ =   
P\_CURRENT\_TXN\_SEQ

UNAPIEV.PARSEEVENTDETAILS

UNAPIEV.FINDDETAILS (SS\_TO, AF, POSTPONED\_AF)

UNAPIEV.EVENTMANAGERJOB('U4EVMGR',**3**);

SELECT

NVL(MAX(DECODE(setting\_name, 'EVMGRS\_EV\_IN\_BULK', setting\_value)),'1') EVMGRS\_EV\_IN\_BULK,

NVL(MAX(DECODE(setting\_name, 'EVMGRS\_POLLING\_ON', setting\_value)),'0') EVMGRS\_POLLING\_ON,

NVL(MAX(DECODE(setting\_name, 'EVMGRS\_POLLINGINTERV', setting\_value)), '450') EVMGRS\_POLLINGINTERV,

NVL(MAX(DECODE(setting\_name, 'EVMGRS\_1QBYINSTANCE', setting\_value)),'0') **EVMGRS\_1QBYINSTANCE**,

NVL(MAX(DECODE(setting\_name, 'EVMGRS\_COLLECTSTAT', setting\_value)),'0') EVMGRS\_COLLECTSTAT,

SYS\_CONTEXT ('USERENV', 'INSTANCE') SESSIONINSTANCENR

FROM utsystem

WHERE setting\_name IN ('EVMGRS\_EV\_IN\_BULK','EVMGRS\_POLLING\_ON','EVMGRS\_POLLINGINTERV','EVMGRS\_1QBYINSTANCE','EVMGRS\_COLLECTSTAT');

-- EVMGRS\_EV\_IN\_BULK EVMGRS\_POLLING\_ON EVMGRS\_POLLINGINTERV EVMGRS\_1QBYINSTANCE EVMGRS\_COLLECTSTAT SESSIONINSTANCENR

-- 0 0 450 0 0 1

UNAPIGEN.SETCONNECTION4INSTALL

UNAPIGEN.SETCONNECTION4INSTALL (incl. timezone)

UTSYSTEM  
DBA\_NAME = UNILAB

JOBS\_DATE\_FORMAT = DDfx/fxMM/RR HH24fx:fxMI:SS

EV\_CLIENT\_ID = ‘EvMgrJob

SERVER = DB

CLIENT\_EVMGR\_USED = YES

Indien **EVMGRS\_1QBYINSTANCE = 0:**

UPDATE UTEV D

SET D.EVMGR\_NAME = 'U4EVMGR'

WHERE TR\_SEQ IN (SELECT C.TR\_SEQ FROM UTEV C WHERE C**.EVMGR\_NAME='1'**)

AND TO\_CHAR(TR\_SEQ) NOT IN (SELECT SUBSTR(NAME,8) TR\_SEQ\_LOCKED

FROM V$LOCK A, SYS.DBMS\_LOCK\_ALLOCATED B

WHERE A.TYPE='UL'

AND A.ID1=B.LOCKID

AND B.NAME LIKE 'U4EVMGR%'

AND B.NAME<> 'U4EVMGRSTART\_OR\_STOP');

Indien EVMGRS\_POLLING\_ON = 0 DAN DBMS\_ALERT.REGISTER(A\_EVMGR\_NAME)

DBMS\_APPLICATION.SET\_ACTION=’Processing events’

CURSOR TR\_SEQ:  
SELECT MIN(TR\_SEQ)FROM UTEV

WHERE (TR\_SEQ >= A\_TR\_SEQ OR TR\_SEQ=-A\_EV\_SESSION)

AND EVMGR\_NAME = A\_EVMGR\_NAME;

DBMS\_LOCK.ALLOCATE\_UNIQUE(‘UAEVMGR||nr) (RETURN: LOCKHANDLE)

Als ik EVMGR=1 geef dan zou hij automatisch moeten worden opgeapkt!!!

**IF NOT L\_EVENT\_PROCESSED then** UNAPIEV3.HANDLEOBJECTEVENTS  
UNAPIEV3.HANDLERQEVENTS  
UNAPIEV3.HANDLESCEVENTS  
UNAPIEV3.HANDLESCICEVENTS  
UNAPIEV3.HANDLESCIIEVENTS  
UNAPIEV4.HANDLESCPGEVENTS  
UNAPIEV4.HANDLESCPAEVENTS  
**UNAPIEV4.HANDLESCMEEVENTS**  
UNAPIEV3.HANDLERQICEVENTS  
UNAPIEV3.HANDLERQIIEVENTS  
UNAPIEV3.HANDLESPECIALEVENTS  
UNAPIEV3.HANDLEEQEVENTS  
UNAPIEV3.HANDLEWSEVENTS  
UNAPIEV3.HANDLECHEVENTS  
UNAPIEV3.HANDLESDEVENTS  
UNAPIEV3.HANDLESDICEVENTS  
UNAPIEV3.HANDLESDIIEVENTS

DBMS\_APPLICATION.SET\_ACTION=’NULL’

**DBMS\_ALERT\_INFO  
NAME MESSAGE  
--------------- ----------  
U4EVMGR 514457  
U4EVMGR 514457  
STUDY\_EVENT\_MGR  
DEDICEVMGR  
U4EVMGR 514457**

UNAPIGEN.ISSYSTEMSUPPORTINGRNDSUITE

Utstystem:  
STARTMEONCELLCREATE = 0

**UNAPIEV4.HANDLESCMEEVENTS**

**UNAPIEV3.HANDLESCEVENTS**

INDIEN P\_EV\_REC.EV\_TP IN   
('SampleCreated',**'SamplePlanned'**)

INDIEN P\_EV\_REC.OBJECT\_TP = ‘sc')

UNAPIEV.P\_EV\_TR = UNAPIEV.P\_EV\_REC

UNAPIEV.UPDATEOPALOBJECTRECORD(UNAPIEV.P\_SS\_TO)

UNACCESS.INITOBJECTACCESSRIGHTS(UNAPIEV.P\_EV\_REC.OBJECT\_TP, UNAPIEV.P\_EV\_REC.OBJECT\_ID, L\_AR)

UNAPIEV.EXECUTETRACTIONS(UNAPIEV.P\_EV\_REC.OBJECT\_SS,UNAPIEV.P\_SS\_TO,UNAPIEV.P\_TR\_NO, L\_TRANSITION\_OK)

UNAPIEV.EVALLIFECYCLE(UNAPIEV.P\_SS\_TO, UNAPIEV.P\_TR\_NO)

DELETEUCAUDITTRAIL('sc', UNAPIEV.P\_SC)

INDIEN P\_EV\_REC.OBJECT\_TP = ‘rq')

UNAPIEV.P\_EV\_TR = UNAPIEV.P\_EV\_REC

UNAPIEV.UPDATEOPALOBJECTRECORD(UNAPIEV.P\_SS\_TO)

UNAPIEV.EVALLIFECYCLE(UNAPIEV.P\_SS\_TO, UNAPIEV.P\_TR\_NO)

UNAPIEV.EXECUTETRACTIONS(UNAPIEV.P\_EV\_REC.OBJECT\_SS,UNAPIEV.P\_SS\_TO,UNAPIEV.P\_TR\_NO, L\_TRANSITION\_OK)

**UNAPIEV3.HANDLESCEVENTS**

**UTEV: INSERT event in UTEV**

UNAPIV.EVALLIFECYCLE

UNAPIEV.EVALLIFECYCLE(UNAPIEV.P\_SS\_TO, UNAPIEV.P\_TR\_NO)

IF P\_EV\_REC.OBJECT\_TP = 'me' THEN UTSCME-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'pa' THEN UTSCPA-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'sdii' THEN UTSDII-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'rqii' THEN UTRQII-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'ii' THEN UTSCII-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'pg' THEN UTSCPG-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'sdic' THEN UTSDIC-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'rqic' THEN UTRQIC-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'ic' THEN UTSCIC-FOR-UPDATE  
**IF P\_EV\_REC.OBJECT\_TP = 'sc' THEN UTSC-FOR-UPDATE !!**  
IF P\_EV\_REC.OBJECT\_TP = 'rq' THEN UTRQ-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'sd' THEN UTSD-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'ws' THEN UTWS-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'ch' THEN UTCH-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'pp' THEN UTPP-FOR-UPDATE  
IF P\_EV\_REC.OBJECT\_TP = 'eq' THEN UTEQ-FOR-UPDATE

Bepaal LC-VERSION

LOOP TR-RECORDS

SELECT SS\_FROM, SS\_TO, TR\_NO, CONDITION

FROM UTLCTR

WHERE LC = P\_EV\_REC.OBJECT\_LC

AND VERSION = P\_EV\_REC.OBJECT\_LC\_VERSION

AND (SS\_FROM = P\_EV\_REC.OBJECT\_SS OR SS\_FROM = '@@')

AND (A\_SS\_TO IS NULL OR SS\_TO = A\_SS\_TO)

ORDER BY SS\_FROM DESC, TR\_NO;

END-LOOP TR-RECORDS

INDIEN CONDITION IS NOT NULL

EXEC UNCONDITION.<loop UTLCTR-condition>

UNAPIV.UPDATEOPALOBJECTRECORD

SELECT ALLOW\_MODIFY, ACTIVE FROM UTSS WHERE SS = A\_SS\_TO

RNDAPIRQP.UpdateRqSoList(:rq);

UNAPIEV.UPDATEOPALOBJECTRECORD(UNAPIEV.P\_SS\_TO)

IF P\_EV\_REC.OBJECT\_TP = 'rq' THEN

UPDATE UTRQ SET LC = P\_EV\_REC.OBJECT\_LC,LC\_VERSION = P\_EV\_REC.OBJECT\_LC\_VERSION, SS = A\_SS\_TO,ALLOW\_MODIFY = L\_ALLOW\_MODIFY, ACTIVE = \_ACTIVE WHERE RQ = P\_RQ;  
INSERT INTO UTRQHS, INSERT INTO UTRQHSDETAILS

IF A\_SS\_TO NOT IN ( '@~', '@P', '@D' ) AND P\_TR\_NO IS NULL THEN L\_SS\_FORCED := TRUE;

IF L\_SS\_FORCED then UNAPIGEN.ISSYSTEM21CFR11COMPLIANT

UNAPIGEN.ISSYSTEMSUPPORTINGRNDSUITE = success?

IF P\_EV\_REC.OBJECT\_TP = 'sc' THEN

UPDATE UTSC SET LC = P\_EV\_REC.OBJECT\_LC,LC\_VERSION = P\_EV\_REC.OBJECT\_LC\_VERSION,SS = A\_SS\_TO,ALLOW\_MODIFY = L\_ALLOW\_MODIFY,ACTIVE = L\_ACTIVEWHERE SC = P\_SC;  
INSERT INTO UTSCHS, INSERT INTO UTSCHSDETAILS

IF L\_SS\_FORCED then UNAPIGEN.ISSYSTEM21CFR11COMPLIANT

IF P\_EV\_REC.OBJECT\_TP = 'sd' THEN

UPDATE UTSD SET LC = P\_EV\_REC.OBJECT\_LC, LC\_VERSION = P\_EV\_REC.OBJECT\_LC\_VERSION, SS = A\_SS\_TO, ALLOW\_MODIFY = L\_ALLOW\_MODIFY, ACTIVE = L\_ACTIVE WHERE SD = P\_SD;  
INSERT INTO UTSDHS, INSERT INTO UTSDHSDETAILS

IF A\_SS\_TO IS NOT NULL

IF A\_SS\_TO IS NULL

**IDEM** VOOR OVERIGE-OBJECT-TP: P\_EV\_REC.OBJECT\_TP IN )'rqic’,’sdic’,’ic’,’rqii’,’sdii’,’pq’,’pa’,’me’,’ws’,’ch’,’

SELECT ALLOW\_MODIFY WHERE SS = P\_EV\_REC.OBJECT\_SS

IF P\_EV\_REC.OBJECT\_TP = 'rq' THEN: UPDATE UTRQ SET ALLOW\_MODIFY = NVL(L\_NEW\_ALLOW\_MODIFY,'0') WHERE RQ = P\_RQ

RETURNING ALLOW\_MODIFY

INTO L\_ALLOW\_MODIFY

IF P\_EV\_REC.OBJECT\_TP = 'sc' THEN: UPDATE UTSC SET ALLOW\_MODIFY = NVL(L\_NEW\_ALLOW\_MODIFY,'0') WHERE SC = P\_SC

RETURNING ALLOW\_MODIFY

INTO L\_ALLOW\_MODIFY

IF P\_EV\_REC.OBJECT\_TP = 'sd' THEN: UPDATE UTSD SET ALLOW\_MODIFY = NVL(L\_NEW\_ALLOW\_MODIFY,'0') WHERE SD = P\_SD

RETURNING ALLOW\_MODIFY

INTO L\_ALLOW\_MODIFY

**IDEM VOOR OVERIGE-OBJECT-TP: P\_EV\_REC.OBJECT\_TP IN )'rqic’,’sdic’,’ic’,’rqii’,’sdii’,’pq’,’pa’,’me’,’ws’,’ch’,’**

IF L\_ALLOW\_MODIFY = '1' THEN   
UNAPIGEN.PA\_OBJECT\_PRIV(L\_SEQ\_NO) := UNAPIGEN.DBERR\_SUCCESS;

UNACCESS.INITOBJECTACCESSRIGHTS

If FOUND then A\_AR = L\_ST\_REC

FETCH l\_inherit\_star

SELECT \* FROM udst WHERE (st, version) =

(SELECT b.st, b.st\_version FROM utsc b WHERE b.sc=a\_sc);

IF L\_AR\_EVENT\_FOUND = false

UNACCESS.INITOBJECTACCESSRIGHTS(UNAPIEV.P\_EV\_REC.OBJECT\_TP, UNAPIEV.P\_EV\_REC.OBJECT\_ID, L\_AR) ‘sc’

IF (a\_object\_tp = 'sc') THEN

ELSIF a\_object\_tp IN ( 'ic', 'rqic', 'sdic' ) THEN

ELSIF a\_object\_tp IN ( 'ii', 'rqii', 'sdii' ) THEN

ELSIF a\_object\_tp IN ( 'pg') THEN

ELSIF a\_object\_tp IN ( 'pa') THEN

ELSIF a\_object\_tp IN ( 'me') THEN

ELSIF a\_object\_tp IN ( 'rg') THEN

ELSIF a\_object\_tp IN ( 'sd') THEN

ELSIF a\_object\_tp IN ( 'ws') THEN

ELSIF a\_object\_tp IN ( 'ch') THEN

SELECT ar FROM utobjects WHERE object = a\_object\_tp;

IF L\_FOUND = false

IF (a\_object\_tp = 'pp') THEN

IF (a\_object\_tp = 'eq') THEN

ELSE

SELECT \* FROM utev

WHERE tr\_seq = UNAPIEV.P\_EV\_REC.tr\_seq

AND object\_tp = UNAPIEV.P\_EV\_REC.object\_tp

AND object\_id = UNAPIEV.P\_EV\_REC.object\_id

AND INSTR(ev\_details, 'version='||UNAPIEV.P\_VERSION) <> 0

AND INSTR(ev\_details, 'lab='||UNAPIEV.P\_LAB) <> 0

AND ev\_tp = 'AccessRightsUpdated';

SELECT \* FROM utev

WHERE tr\_seq = UNAPIEV.P\_EV\_REC.tr\_seq

AND object\_tp = UNAPIEV.P\_EV\_REC.object\_tp

AND object\_id = UNAPIEV.P\_EV\_REC.object\_id

AND INSTR(ev\_details, 'version='||UNAPIEV.P\_VERSION) <> 0

AND ev\_tp = 'AccessRightsUpdated';

SELECT \*FROM utev

WHERE tr\_seq = UNAPIEV.P\_EV\_REC.tr\_seq

AND object\_tp = UNAPIEV.P\_EV\_REC.object\_tp

AND object\_id = UNAPIEV.P\_EV\_REC.object\_id

AND INSTR(ev\_details, 'version='||UNAPIEV.P\_VERSION) <> 0

AND INSTR(ev\_details, 'pp\_key1='||UNAPIEV.P\_PP\_KEY1) <> 0

AND INSTR(ev\_details, 'pp\_key2='||UNAPIEV.P\_PP\_KEY2) <> 0

AND INSTR(ev\_details, 'pp\_key3='||UNAPIEV.P\_PP\_KEY3) <> 0

AND INSTR(ev\_details, 'pp\_key4='||UNAPIEV.P\_PP\_KEY4) <> 0

AND INSTR(ev\_details, 'pp\_key5='||UNAPIEV.P\_PP\_KEY5) <> 0

AND ev\_tp = 'AccessRightsUpdated';

UNAPIEV.EXECUTETRACTIONS

L\_EV\_SEQ\_NR := -1;

L\_EV\_DETAILS := UNAPIEV.P\_EV\_REC.EV\_DETAILS ||

'#af='||L\_AF.AF||

'#postpone\_af='||TO\_CHAR(UNAPIGEN.DBERR\_TRANSITION);

L\_RESULT := UNAPIEV.INSERTEVENT(UNAPIEV.P\_EV\_REC.DBAPI\_NAME, UNAPIGEN.P\_EVMGR\_NAME,

UNAPIEV.P\_EV\_REC.OBJECT\_TP, UNAPIEV.P\_EV\_REC.OBJECT\_ID,

UNAPIEV.P\_EV\_REC.OBJECT\_LC, UNAPIEV.P\_EV\_REC.OBJECT\_LC\_VERSION,

UNAPIEV.P\_EV\_REC.OBJECT\_SS, UNAPIEV.P\_EV\_REC.EV\_TP,

L\_EV\_DETAILS, L\_EV\_SEQ\_NR);

Indien ERR + POSTPONED-AF = 0

LOOP AF-RECORDS

SELECT AF, SS\_FROM

FROM UTLCAF

WHERE LC = P\_EV\_REC.OBJECT\_LC

AND VERSION = P\_EV\_REC.OBJECT\_LC\_VERSION

AND (SS\_FROM = NVL(P\_LC\_SS\_FROM,A\_SS\_FROM) OR SS\_FROM = '@@')

AND TR\_NO = A\_TR\_NO

AND SS\_TO = A\_SS\_TO

ORDER BY SS\_FROM DESC, SEQ;

END-LOOP AF-RECORDS

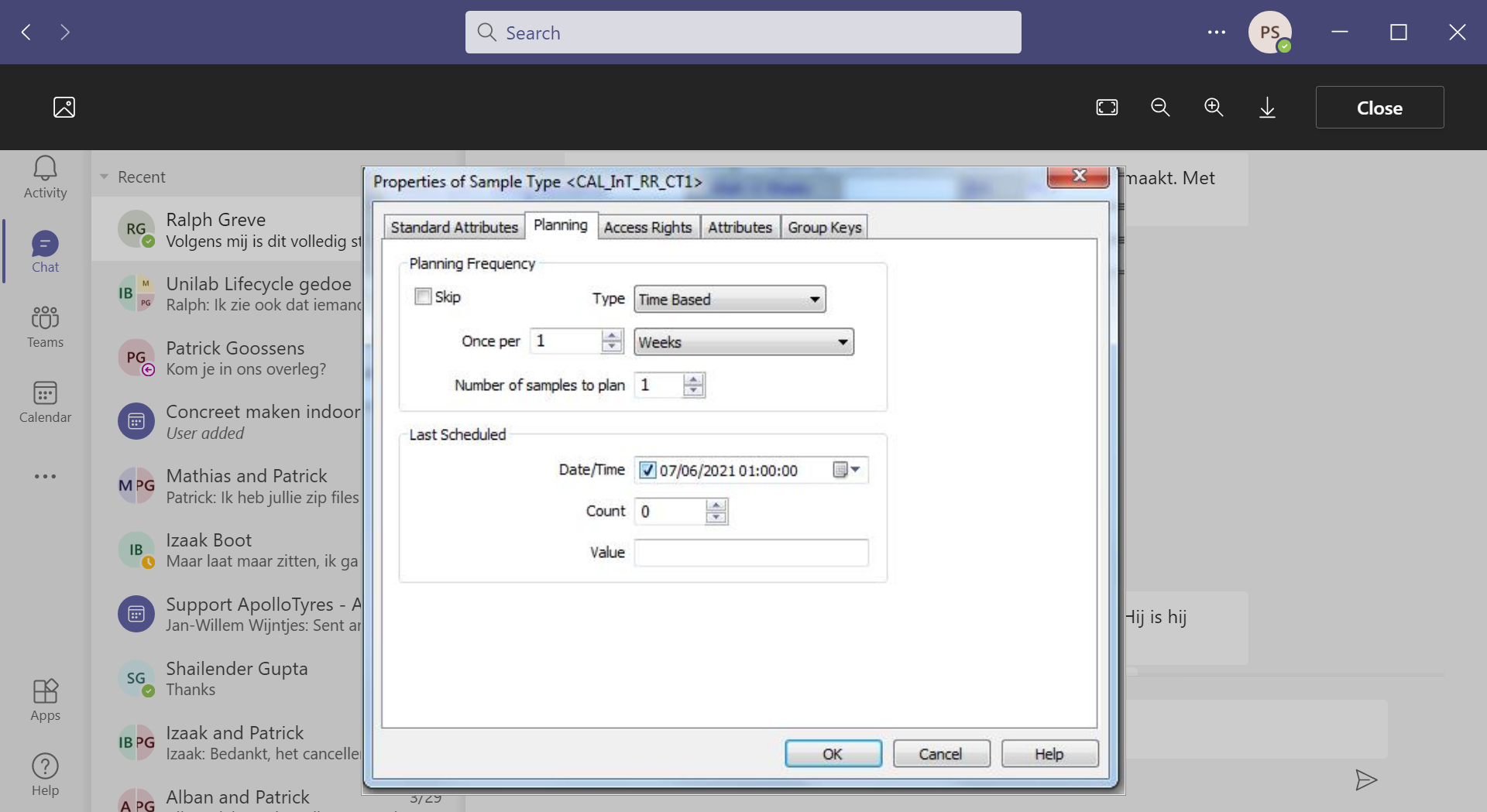
EXEC UNACTION.<loop UTLCAF-AF>

UNAPIEV.EXECUTETRACTIONS(UNAPIEV.P\_EV\_REC.OBJECT\_SS,UNAPIEV.P\_SS\_TO,UNAPIEV.P\_TR\_NO, L\_TRANSITION\_OK)

Action postponed to the end of the action queue

Configuratie UNILAB tbv SAMPLE-PLANNING

Ga naar SampleType-properties



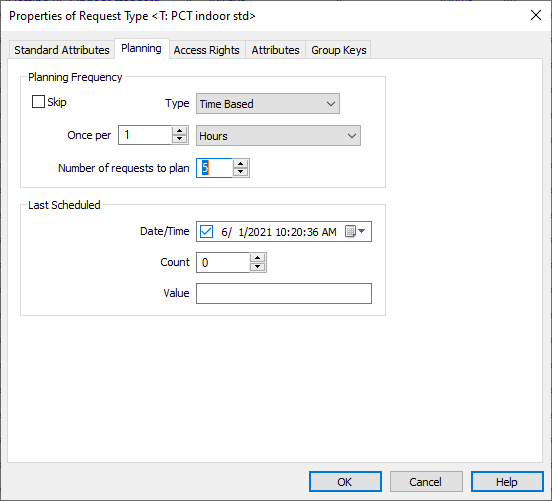
Job maakt wekelijks (maandag-ochtend 01:00:00 uur) samples aan, let op: ZONDER een REQUEST !!.

Deze JOB draait maar 1/wk, dus het opgeven van een interval KLEINER dan 1 wk heeft helemaal geen NUT !!.

LET OP:

Job werkt alleen op SAMPLES en niet op REQUESTS !!! (komt ook wel overeen met naamgeving van de job: SamplePlanner en package: UNAPIPLAN.SamplePlanner)

Wel vreemd op REQUEST-niveau kun je wel FREQUENTIE etc. configureren, maar er wordt dus blijkbaar niets mee gedaan !



-hoofd-query binnen SamplePlanner is ook alleen over UTSC  
-system-parameters bestaan alleen voor SC, en niet voor RQ, waarmee een evt. request-planner-proces aangestuurd zou kunnen worden.   
-binnen create-samples procedure worden nergens REQUESTS aangemaakt. In huidige SamplePlanner kunnen dus technisch ook alleen maar SAMPLES worden aangemaakt.