Naamgeving Customisaties:

AT = ADDON TABLES  
AP = ADDON PACKAGES

Client logging: <Installatie-folder>\unerror – logfile

Database-logging: Uitvragen aansturing via SYSTEM-settings:

Select \* from utsystem where setting\_name like 'LOG%';

ORACLEPROD

LOG NO gebruik vanuit unfreq.UabasedSt/UabasedSc, input voor TraceError!

LOG\_EV 0 gebruik vanuit unapigen. SETCONNECTION4INSTALL (1=p\_log\_ev=true)

LOG\_INFO 0 gebruik vanuit apaofunctions.loginfo

LOG\_LC\_ACTIONS 0 gebruik vanuit unapigen. SETCONNECTION4INSTALL (1=p\_log\_lc\_actions=true)

LOG\_OUTDOOR\_LVL 1 gebruik vanuit apao\_outdoor.LogError

Log-tabel: UTEVLOG:

UTSYSTEM: event-logging via TABEL UTEVLOG vullen indien setting is changed van 0 naar 1.  
 Setting: LOG\_EV, LOV\_LC\_ACTIONS

Log-tabel: UTERROR:

Basis-functie om log-regels weg te schrijven vanuit UNILAB:

UNAPIGEN.LogError (lcs\_function\_name, 'Save1WsScRow returned ' ||

lvi\_ret\_code || ' for ws <' || lvs\_ws || '>');

Custom-functies die ook log-regels wegschrijven:

UTSYSTEM: apaofunctions: setting LOG\_INFO changen 0 naar 1.  
 Vanuit procedure: Apaogen.logerror   
 APAOFUNCTIONS.LogInfo(lcs\_function\_name, 'WS\_A01:' || lcs\_function\_name ||  
 ' Before CREATEWS for ' || '#WT=' || lvr\_wt.wt ||'#PR=' || lvr\_wt.pr );

UTSYSTEM: apao\_outdoor: setting LOG\_OUTDOOR\_LVL wijzigen naar 1,2,3,4   
 (voor resp. 1=ERROR, 2-=WARN, 3=INFO, 4=DEBUG-meldingen)  
 Vanuit procedure: Outdoor.LogError

**GEBRUIK UNAPIGEN.LOGERROR**

PROCEDURE LOGERROR (A\_API\_NAME IN VARCHAR2,

A\_ERROR\_MSG IN VARCHAR2)

IS

BEGIN

IF UNAPIEV.P\_EV\_MGR\_SESSION AND

SQLCODE = -60 AND

UNAPIGEN.P\_DEADLOCK\_COUNT < UNAPIGEN.P\_MAX\_DEADLOCK\_COUNT THEN

UNAPIGEN.P\_DEADLOCK\_RAISED := TRUE;

ELSE

IF **P\_REMOTE** = '0' THEN

LOGERRORNOREMOTEAPICALL(A\_API\_NAME, A\_ERROR\_MSG);

ELSE

INSERT INTO **UTERROR**(CLIENT\_ID, APPLIC, WHO, LOGDATE, LOGDATE\_TZ, API\_NAME, ERROR\_MSG)

VALUES (UNAPIGEN.P\_CLIENT\_ID, SUBSTR(UNAPIGEN.P\_APPLIC\_NAME,1,8), NVL(UNAPIGEN.P\_USER,USER),  
 CURRENT\_TIMESTAMP, CURRENT\_TIMESTAMP, SUBSTR(A\_API\_NAME,1,40), SUBSTR(A\_ERROR\_MSG,1,255));

END IF;

IF SQLCODE = -60 THEN

UNAPIGEN.P\_DEADLOCK\_RAISED := TRUE;

END IF;

END IF;

UNAPIGEN.P\_TXN\_ERROR\_TEXT := SUBSTR(A\_ERROR\_MSG,1,255);

IF UNAPIGEN.P\_TXN\_ERROR = UNAPIGEN.DBERR\_SUCCESS THEN

UNAPIGEN.P\_TXN\_ERROR := UNAPIGEN.DBERR\_GENFAIL;

END IF;

END LOGERROR;

PROCEDURE LOGERRORNOREMOTEAPICALL

(A\_API\_NAME IN VARCHAR2,

A\_ERROR\_MSG IN VARCHAR2)

IS

**PRAGMA AUTONOMOUS\_TRANSACTION;**

BEGIN

INSERT INTO **UTERROR**(CLIENT\_ID, APPLIC, WHO, LOGDATE, LOGDATE\_TZ, API\_NAME, ERROR\_MSG)

VALUES (UNAPIGEN.P\_CLIENT\_ID, SUBSTR(UNAPIGEN.P\_APPLIC\_NAME,1,8), NVL(UNAPIGEN.P\_USER,USER),  
 CURRENT\_TIMESTAMP, CURRENT\_TIMESTAMP, SUBSTR(A\_API\_NAME,1,40), SUBSTR(A\_ERROR\_MSG,1,255));

COMMIT;

END LOGERRORNOREMOTEAPICALL;

**CONCLUSIES:**-Indien REMOTE=1 dan GEEN autonomous-transaction, maar dan ook GEEN COMMIT !!!

-indien aangeroepen vanuit UNAPI-packages dan wordt P\_REMOTE=1 gezet voor een INTERNAL-UPDATE van bijv. PA, etc. Na de verwerking krijgt P\_REMOTE zijn oorspronkelijke waarde (meestal=0) weer terug.

-

Gebruik APAOGEN.LogError  
**Package Aanroep**Apao\_outdoor allen zonder, dus via rollback en via UNAPIGEN.LogError  
ApaoCondition allen zonder, dus via rollback en via UNAPIGEN.LogError  
ApaoEvRules allen zonder, alleen 1x **meCustomSQL.logerror** met rollback=FALSE !  
ApaoFunctions allen zonder, alleen 1x **loginfo.logerror** met rollback/commit=FALSE !  
ApaoRegularRelease allen zonder, dus via rollback en via UNAPIGEN.LogError  
ApaoTrials allen zonder, dus via rollback en via UNAPIGEN.LogError  
At\_WMS allen zonder, dus via rollback en via UNAPIGEN.LogError  
UnAction allen zonder, dus via rollback en via UNAPIGEN.LogError  
UnCondition allen zonder, dus via rollback en via UNAPIGEN.LogError

**GEBRUIK VAN APAOGEN.LOGERROR**

PROCEDURE LogError (avs\_function\_name IN API\_NAME\_TYPE,

avs\_message IN VARCHAR2,

avb\_with\_rollback IN BOOLEAN := TRUE,

avb\_with\_commit IN BOOLEAN := TRUE,

avi\_message\_length IN INTEGER := 255)

IS

lcs\_function\_name CONSTANT API\_NAME\_TYPE := ics\_package\_name||'.'||'LogError';

lvs\_sqlerrm ERROR\_MSG\_TYPE;

lvi\_ret\_code RETURN\_TYPE;

lvs\_line\_left VARCHAR2 (32767) := avs\_message;

lvs\_line uterror.error\_msg%TYPE;

lvi\_counter INTEGER;

lvs\_line\_bak uterror.error\_msg%TYPE;

lvs\_line\_left\_bak VARCHAR2 (32767);

PROCEDURE WriteLine (avs\_line VARCHAR2) IS

BEGIN

IF avb\_with\_rollback THEN

UNAPIGEN.LogError (avs\_function\_name, **avs\_line**);

ELSE

INSERT INTO uterror(client\_id, applic, who, logdate, api\_name, error\_msg)

VALUES(UNAPIGEN.P\_CLIENT\_ID, UNAPIGEN.P\_APPLIC\_NAME, UNAPIGEN.P\_USER, CURRENT\_TIMESTAMP,

avs\_function\_name, avs\_line);

IF avb\_with\_commit THEN

COMMIT;

END IF;

END IF;

END WriteLine;

BEGIN

WHILE LENGTH (lvs\_line\_left) > avi\_message\_length LOOP

**lvs\_line** := SUBSTR (lvs\_line\_left, 1, avi\_message\_length);

lvs\_line\_left := SUBSTR (lvs\_line\_left, avi\_message\_length + 1);

lvi\_counter := -1;

lvs\_line\_bak := lvs\_line;

lvs\_line\_left\_bak := lvs\_line\_left;

WHILE INSTR (ics\_break\_chars, SUBSTR (lvs\_line, -1)) <= 0 LOOP

lvi\_counter := lvi\_counter + 1;

lvs\_line\_left := SUBSTR (lvs\_line, -1)||lvs\_line\_left;

lvs\_line := SUBSTR (lvs\_line, 1, LENGTH (lvs\_line) - 1);

IF lvs\_line IS NULL THEN

lvs\_line := lvs\_line\_bak;

lvs\_line\_left := lvs\_line\_left\_bak;

EXIT;

END IF;

END LOOP;

WriteLine (**lvs\_line**);

END LOOP;

WriteLine (lvs\_line\_left);

EXCEPTION

WHEN OTHERS THEN

IF SQLCODE != 1 THEN LogError (lcs\_function\_name, SQLERRM);

END IF;

END LogError;

**Conclusies:**

-**logError gebruikt om lange string op te knippen in stukken van 255 karakters** !!!

-Geen AUTONOMOUS-TRANSACTION indien ROLLBACK=FALSE + COMMIT=TRUE!!!!  
-Indien rollback/commit=false, dan wel een uterror-INSERT, maar geen commit of ROLLBACK !!!  
-In veel situaties zou aanroep door UNAPIGEN.LOGERROR vervangen kunnen worden !!

**Gebruik SYSTEM-SETTING = LOG**

**UNFREQ**

PROCEDURE TraceError

(a\_api\_name IN VARCHAR2, /\* VC40\_TYPE \*/

a\_log\_error IN VARCHAR2, /\* HVB3\*/

a\_key IN VARCHAR2, /\* HVB3\*/

a\_seq IN OUT NUMBER,

a\_error\_msg IN VARCHAR2 /\* VC255\_TYPE \*/

) /\* HVB3\*/

IS

**PRAGMA AUTONOMOUS\_TRANSACTION;**

lvi\_error\_msg VARCHAR2(255);

BEGIN

--HVB3 IF APAOGEN.GetSystemSetting('LOG','NO') = 'YES' THEN

IF a\_log\_error= 'YES' THEN --HVB3

lvi\_error\_msg := a\_key || TO\_CHAR(a\_seq) || ':' ||a\_error\_msg;

--autonomous transaction used here

--UNAPIGEN.LogError is also an autonomous transaction but may rollback the current transaction

INSERT INTO **uterror**(client\_id, applic, who, logdate, api\_name, error\_msg)

VALUES (UNAPIGEN.P\_CLIENT\_ID, SUBSTR(UNAPIGEN.P\_APPLIC\_NAME,1,8), NVL(UNAPIGEN.P\_USER,USER),   
 CURRENT\_TIMESTAMP, SUBSTR(a\_api\_name,1,40), SUBSTR(lvi\_error\_msg,1,255));

COMMIT;

a\_seq := a\_seq + 1; --HVB3

END IF;

END TraceError;

**Gebruik SYSTEM-SETTING = LOG\_OUTDOOR\_LVL**

**APAO\_OUTDOOR**

PROCEDURE LogError(avs\_func\_name IN VARCHAR2,

avs\_msg IN VARCHAR2,

avn\_lvl IN INTEGER default 4) --0=no\_logging, 1=error, 2=warning, 3=info, 4=debug

IS

lvn\_uterror\_seq INTEGER;

lvn\_log\_lvl NUMBER;

lvs\_log\_lvl\_descr VARCHAR2(6);

BEGIN

lvn\_log\_lvl:=TO\_NUMBER(APAOGEN.GETSYSTEMSETTING('LOG\_OUTDOOR\_LVL', 0));

IF avn\_lvl <= lvn\_log\_lvl THEN --check whether to log this message

SELECT UTERROR\_SEQ.Nextval

INTO lvn\_uterror\_seq

from dual;

CASE avn\_lvl

WHEN 1 THEN lvs\_log\_lvl\_descr:='ERROR';

WHEN 2 THEN lvs\_log\_lvl\_descr:='WARN';

WHEN 3 THEN lvs\_log\_lvl\_descr:='INFO';

WHEN 4 THEN lvs\_log\_lvl\_descr:='DEBUG';

ELSE lvs\_log\_lvl\_descr:='UNKNOWN LOG LEVEL';

END CASE;

UNAPIGEN.LogError ( avs\_func\_name, LPAD(lvn\_uterror\_seq, 4, '0') || ' - ' || lvs\_log\_lvl\_descr || ' - '|| avs\_msg );

END IF;

END LogError;

Conclusies:

-LEVEL wordt wel doorgegeven aan UNAPIGEN.LOGERROR maar daar wordt vervolgens NIETS mee gedaan, qua FILTERING, alleen komt er een TEKST "ERROR/WARN/INFO/DEBUG" voor te staan. Dat schiet niet op…

Gebruik van SYSTEM-SETTING = LOG\_INFO

APAOFUNCTIONS

Overige LOG-functions:

Gebruik UNAPIRAAT.LogError

Gebruik UNAPIRAAU.LogError

Gebruik UNAPIRACH.LogError

Gebruik UNAPIRACY.LogError

Gebruik UNAPIRAEQ.LogError

Gebruik UNAPIRAGK.LogError

Gebruik UNAPIRAIE.LogError

Gebruik UNAPIRAIP.LogError

Gebruik UNAPIRAJO.LogError

Gebruik UNAPIRALC.LogError

Gebruik UNAPIRALY.LogError

Gebruik UNAPIRAMT.LogError

Gebruik UNAPIRAPP.LogError

Gebruik UNAPIRAPR.LogError

Gebruik UNAPIRAPT.LogError

Gebruik UNAPIRARQ.LogError

Gebruik UNAPIRART.LogError

Gebruik UNAPIRASC.LogError

Gebruik UNAPIRASD.LogError

Gebruik UNAPIRAST.LogError

Gebruik UNAPIRAT3.LogError

Gebruik UNAPIRANS.LogError

Gebruik UNAPIRAWT.LogError

Package: UN\_TRACE = pipe output naar tabel/sqlplus/database etc.

Schrijft output in table = UTEVTRACE

Hoe aanzetten?