



**ZBATCH - taustaprotsessid**

**Digiretsept - arendustööde spetsifikatsioon**



## SISUKORD

<b>1 Üldine</b>	<b>4</b>
1.1 Spetsifikatsiooni eesmärk ja skoop	4
1.2 Arendustöö eesmärk	4
1.3 Spetsifitseeritavad protsessid	4
<b>2 Taustaprotsessid</b>	<b>4</b>
2.1 Andmetabelid	4
2.1.1 ZDR_BATCH struktuur	5
2.1.2 ZDM_BATCH_TR struktuur	5
2.1.3 ZDT_BATCH_LOG struktuur	5
2.2 Erisoodustuste limiitide uuendused	6
2.3 Meditsiiniseadme kaartide limiitide uuendused	7
2.4 Apteegis broneeritud retseptide staatuse muutmine	8
2.5 Tähtaja ületanud retseptide annulleerimine	9
<b>3 Logifunktsioon</b>	<b>9</b>

### Versioonihaldus

Versioon	Versiooni / muutuse kirjeldus	Kuupäev	Autor
0.1	Esimene versioon	27.02.2008	Indrek Vaino
0.2	Täiendatud ja parandatud versioon: uuendusperioodide ja broneeringu tühistamise erinev käsitlus	16.04.2008	Indrek Vaino

# 1 Üldine

## 1.1 Spetsifikatsiooni eesmärk ja skoop

Käesoleva dokumendi eesmärk on spetsifitseerida taustaprotsesside töötlemise arenduse ZBATCH funktsionaalsus. Käesoleva spetsifikatsiooni skoopi ei kuulu andmesiirde liideste kirjeldused, kuna need on R/3-e ehk retseptikeskuse tuuma jaoks väljastpoolt käivitatavad teenused, mille käivituse aja ning sageduse määrab nende välja kutsuja (XI). Nende alla kuulub ka patsiendi surma faktiga seotud tegevuste läbiviimine, kuna surma kuupäev on R/3-e jaoks väline parameeter. Kõik BW-ga seotud andmevahetuse toimingud kuuluvad BW konfiguratsiooni dokumenti (seadistused andmete pärimiseks tehakse BW-s ning R/3-e jaoks on tegu pull-toimingutega).

## 1.2 Arendustöö eesmärk

Käesoleva dokumendiga kirjeldatava taustaprotsesside lahenduse eesmärk on vastavalt etteantud parameetritele teostada süsteemis teatud regulaarseid taustal läbiviidavaid toiminguid – taustaprotsesse ning logida teostatud toimingud hilisemaks monitooringuks.

## 1.3 Spetsifitseeritavad protsessid

Taustaprotsessideks retseptikeskuse mõistes on järgmiste protsesside läbiviimine:

- erisoodustuse limiitide uuendusprotseduur,
- meditsiiniseadme kaartide limiitide uuendusprotseduur,
- apteegis broneeritud retseptide tagasikeeramine staatusesse „välja kirjutatud”,
- tähtaja ületanud retseptidele staatuse „annulleeritud” omistamine.

# 2 Taustaprotsessid

Taustaprotsesside jaoks tuleb luua tavaline scheduler'iga käivitatav programm ZDRR\_BATCH. Peab olema välistatud, et seda käivitataks käsitsi reaalsajas.

## 2.1 Andmetabelid

Käesoleva spetsifikatsiooni raames kasutatakse järgmisi andmetabeleid.

Tabel	Nimetus
ZDM_ERISOODUD	Erisoodustuste register.
EKKO	Meditsiiniseadme kaartide register / retseptide register.
ZDR_BATCH	Taustaprotsesside seadistusparameetrid.
ZDR_LIMIT	MSk / erisoodustuste limiidiuuenduste koodide tabel.
ZDT_BATCH_LOG	Taustaprotsesside logitabel.
ZDM_BATCH_TR	Taustaprotsesside toimingute koodide tabel.

### 2.1.1 ZDR\_BATCH struktuur

Taustaprotsesside seadistusparameetrite tabelil on alljärgnev struktuur.

Väli	Andmeelement	Kirjeldus
MANDT	MANDT	Client.
ZLOGACT	CHAR (1)	Kas logimise funktsioon on aktiivne? (X / initial)
ZERIS	CHAR (1)	Kas uuendada erisoodustuste limiite? (X / initial)
ZERISPER	ZDR_LIMIT-ZCODE	Erisoodustuse limiitide uuendusperiood.
ZLE_DATE	DATS	Viimane erisoodustuse limiitide uuendamise kuupäev.
ZMSK	CHAR (1)	Kas uuendada meditsiiniseadme kaartide limiite? (X / initial)
ZMSK_DATE	DATS	Viimane meditsiiniseadmete limiitide uuendamise kuupäev
ZBRON	CHAR (1)	Kas muuta broneeritud retseptide staatust? (X / initial)
ZBRST_DAYS	INT1	Broneeritud staatuse tagasikeeramise aeg päevades.
ZBR_DATE	DATS	Viimane broneeringute staatuste tagasikeeramise kuupäev.
ZANN	CHAR (1)	Annulleerida tähtaja ületanud retseptid? (X / initial)
ZANN_DATE	DATS	Viimane annulleerimise kuupäev.

Unikaalne indeks: MANDT.

Tabel peab olema sisestatav ning andmete muudatusi tuleb logida. Väljad ZLE\_DATE, ZMSK\_DATE, ZBR\_DATE, ZANN\_DATE pole kasutaja jaoks muudetavad vaid read only.

CHAR (1) tüüpi väljad on checkboxid.

Peavad kehtima tingimused

- ZERIS AND ZERISPER == 1.
- ~~ZBRON AND ZBRST\_DAYS == 1 (kusjuures 1 < ZBRST\_DAYS < 365).~~

### 2.1.2 ZDM\_BATCH\_TR struktuur

ZDM\_BATCH\_TR tabelis hoitakse taustaprotsesside toimingute koode ning kirjeldusi.

Väli	Andmeelement	Kirjeldus
MANDT	MANDT	Client.
ZTR	ZZBATCHTR (CH: 2)	Toimingu kood järgmiselt: 10 – erisoodustuste limiitide uuendamine, 20 – meditsiiniseadmete limiitide uuendamine, 30 – broneeringu mahakeeramine tähtaja ületamisel, 40 – annulleerimine tähtaja ületamisel.
ZTX	ZZLTX	Toimingu tüübile vastav tekst.

Unikaalne indeks: MANDT, ZTR.

Tabelis hoitakse loendit eeldefineeritud taustaprotsesside toimingutüüpidest ning see on mõeldud muutmiseks üksnes arendaja poolt.

### 2.1.3 ZDT\_BATCH\_LOG struktuur

ZDT\_BATCH\_LOG tabelis logitakse taustaprotsesside käigus teostatud toiminguid.

Väli	Andmeelement	Kirjeldus
MANDT	MANDT	Client.
ZTR	ZZBATCHTR	Toimingu kood.

ZDATS	DATS	Toimingu läbiviimise kuupäev.
ZTIMS	TIMS	Toimingu läbiviimise kellaaeg.
ZPARAM1	CHAR (20)	Toimingu väljundparameeter 1.
ZPARAM2	CHAR (20)	Toimingu väljundparameeter 2.
ZVAL1	CHAR (20)	Toimingu parameetri väärtus 1.
ZVAL2	CHAR (20)	Toimingu parameetri väärtus 2.
ZVAL3	CHAR (20)	Toimingu parameetri väärtus 3.

Unikaalne indeks: MANDT, ZTR, ZDAST, ZTIMS, ZPARAM1, ZPARAM2, ZVAL1.

## 2.2 Erisoodustuste limiitide uuendused

Leiame uuendusperioodi alguskuupäeva:

- &UP\_DATE& ← kalendriaasta esimene kuupäev, kui ZDR\_BATCH-ZERISPER = „12” (nt. 01.01.2009, 01.01.2012 jne),
- &UP\_DATE& ← kalendri poolaasta esimene kuupäev, kui ZDR\_BATCH-ZERISPER = „06” (nt. 01.07.2009, 01.07.2012 jne).

Leiame uuendusperioodi lõppkuupäeva:

&UP\_END\_DATE& ← viimane kuupäev &UP\_DATE& kuu väärtusest (nt. &UP\_DATE& == 01.01.2009 ? &UP\_END\_DATE& ← 31.01.2009).

Sündmuse toimumise kontrollitingimused.

- &UP\_DATE& LE today() LE &UP\_END\_DATE& (uuendame ühe kuu jooksul vahemikus),
- ZDM\_ERISOODUD-ZLUPDATE LT &UP\_DATE& (selle perioodi uuendamata jäänud erisoodustuse limiite),
- ZDR\_BATCH-ZERIS != INITIAL (juhul, kui seadistustes on see nõutud).

Limiitide uuendamise reeglid.

Väärtustada väljad tabelis ZDM\_ERISOODUD järgmiselt

- ZRES\_AMT ← ZLIM\_AMT,
- ZRES\_QTY ← ZLIM\_QTY,
- ZLUPDATE ← today(),

järgmistel ridadel:

- ZLOPP > today() AND
- ZALGUS < toda() AND
- ZACT IS „YES”.

Iga väärtustatud rea kohta kutsuda välja funktsiooni &\_ZLOG& järgmiste parameetrite väärtustega (juhul, kui ZDR\_BATCH-ZLOGACT != INITIAL):

- &TR\_TYPE& ← „10”,
- &TR\_DATE& ← today(),
- &TR\_TIME& ← jooksev kellaaeg,
- &TR\_PARM1& ← ZDM\_ERISOODUD-MATNR,
- &TR\_PARM2& ← ZDM\_ERISOODUD-ZPATSIKOOD,

- &TR\_VAL1& ← ZDM\_ERISOODUD-ZLIM\_AMT,
- &TR\_VAL2& ← ZDM\_ERISOODUD-ZLIM\_QTY.

Omistada ZDR\_BATCH-ZLE\_DATE väärtus today().

## 2.3 Meditsiiniseadme kaartide limiitide uuendused

### Limiidide uuendusreeglid ja perioodi muutujate omistamine.

ZDR\_BATCH-ZMSK != INITIAL ning today() jääb ajavahemikku

- Kvartali esimene kuupäev kuni kvartali esimese kuu viimane kuupäev >> &UP\_DATE& ← kvartali esimene kuupäev ning &UP\_END\_DATE& ← kvartali esimese kuu viimane kuupäev? (uuendustüüp 03),
- Poolaasta esimene kuupäev poolaasta esimese kuu viimane kuupäev >> &UP\_DATE& ← poolaasta esimene kuupäev ning &UP\_END\_DATE& ← poolaasta esimese kuu viimane kuupäev? (uuendustüüp 06),
- Kalendriaasta esimene kuupäev kalendriaasta esimese kuu viimane kuupäev >> &UP\_DATE& ← kalendriaasta esimene kuupäev ning &UP\_END\_DATE& ← kalendriaasta esimese kuu viimane kuupäev? (uuendustüüp 12).

Kui mingi ülaltoodust sai vastuse „Jah” (nt. today() eq 31.01.2009 või 15.07.2010 jne), siis jätkame punktidega (i), (ii) ja (iii).

(i) Leiame EKKO kirje, milles

- LIFNR EQ „MS\_KAART” ning
- BSTYP EQ „K” (Contract) ning
- BSART EQ „MK” ning
- IHRAN GT &UP\_END\_DATE& ~~today()~~ (quotation date) ning
- BEDAT LT &UP\_DATE& ~~today()~~ ning
- ZZSTATUS EQ „00” (välja kirjutatud, mitte annulleeritud).

Uuendada leitud EKKO kirjet järgmiselt:

a) juhul, kui ZZMSKUL EQ „03” ning today() on kvartali esimene kuupäev, siis

- o KDATB ← today(),
- o KDATE ← kvartali lõppkuupäev,
- o Leiame EKKO-le vastava(d) EKPO read, ning omistame EKPO-KTMNG ← EKKO-UPINC;

b) juhul, kui ZZMSKUL EQ „06” ning today() on poolaasta esimene kuupäev, siis

- o KDATB ← today(),
- o KDATE ← poolaasta lõpu kuupäev,
- o Leiame EKKO-le vastava(d) EKPO read, ning omistame EKPO-KTMNG ← EKKO-UPINC;

c) juhul, kui ZZMSKUL EQ „12” ning today() on kalendriaasta esimene kuupäev, siis

- a. KDATB ← today(),
- b. KDATE ← aasta lõpu kuupäev,

- c. Leiame EKKO-le vastava(d) EKPO read, ning omistame EKPO-KTMNG ← EKKO-UPINC.

(ii) Annulleerime olemasoleva MSk-ga seotud PO.

Siin peame skooopi võtma veel ühe EKKO tabeli (EKKO\_1, ning vana on EKKO\_old).

Leiame EKKO\_1 tabelist kirje nii, et

- EKKO\_1-ZZSTATUS EQ „00” (välja kirjutatud, müüdav) ning
- EKKO\_1-KONNR EQ EKKO\_old-EBELN või EKKO\_1-KUNNR EQ EKKO\_old-EBELN.

Leitud kirjel uuendame staatust: EKKO\_1-ZZSTATUS ← „99” (annulleeritud) ning loome uue MSk-ga seotud retsepti kasutades selleks ZCORE spetsifikatsioonis alapunktis 11.3 toodud MSk retsepti loomise kirjeldust. Vajalikud struktuurid võib täita, kutsudes enne välja BAPI\_PO\_GETDETAIL1 (kasutades sisendina PO numbrit EKKO\_1-EBELN).

(III) Toimingu logimine (juhul, kui ZDR\_BATCH-ZLOGACT != INITIAL).

Iga väärtustatud rea kohta kutsuda välja funktsiooni &\_ZLOG& järgmiste parameetrite väärtustega:

- &TR\_TYPE& ← „20”,
- &TR\_DATE& ← today(),
- &TR\_TIME& ← jooksev kellaaeg,
- &TR\_PARM1& ← EKKO\_old-EBELN (Contract number),
- &TR\_PARM2& ← EKKO\_1-EBELN (PO number),
- &TR\_VAL1& ← EKKO\_old-KDATB (uus kehtivuse alguskuupäev),
- &TR\_VAL2& ← EKKO\_old-KDATE (uus kehtivuse lõpukuupäev),
- &TR\_VAL3& ← EKKO\_old-UPINC (MSk limiit).

(iv) Leiame järgmise meditsiiniseadme kaardi (tagasi punkti (i)).

Omistada ZDR\_BATCH-ZMSK\_DATE väärtus today().

## 2.4 Apteegis broneeritud retseptide staatuse muutmine

(Viimane) broneerimise kuupäev ja kellaaeg on salvestatud retseptile välja EKKO-ZZDATS ning EKKO-ZZTIMS, antud protseduuri käigus tuleb kasutada kontrolliks nende väljade väärtusi.

### Broneeringu mahavõtmise tingimused.

ZDR\_BATCH-ZBRON != INITIAL.

Viia läbi broneeringu tühistamise toiming vastavalt spetsifikatsioonis ZCORE peatükile 8 (Retsepti broneeringu tühistamine) järgmiste PO-de korral

- EKKO-ZZSTATUS EQ „20” (broneeritud apteegis),
- ( ZDR\_BRONX-ZKEHTT + ZDR\_BRONX-ZKEHTM ( WHERE ZDR\_BRONX-BSART EQ EKKO-BSART ) < EKKO-ZZTIMS ) ? &DATEI& ← 1 : &DATEI& = 0 (ehk siis kui bronni aeg + kehtivuse aeg tundides ning minutites ületab täiskuupäeva, siis on kehtivuse lõpukuupäevaks järgmine kuupäev),
- Today() >= EKKO-ZZDATS + &DATEI& ( WHERE EKKO-BSART EQ ZDR\_BRONX-BSART ).

Iga väärtustatud rea kohta kutsuda välja funktsiooni &\_ZLOG& järgmiste parameetrite väärtustega (juhul, kui ZDR\_BATCH-ZLOGACT != INITIAL):

- &TR\_TYPE& ← „30”,

- &TR\_DATE& ← today(),
- &TR\_TIME& ← jooksev kellaaeg,
- &TR\_PARM1& ← EKKO-EBELN,
- &TR\_PARM2& ← EKKO-BEDAT,
- &TR\_VAL1& ← &UP\_DATS&,
- &TR\_VAL2& ← „20” (staatus: broneeritud apteegis),
- &TR\_VAL3& ← „00” (staatus: müügikõlbulik).

Omistada ZDR\_BATCH-ZBR\_DATE väärtus today().

## 2.5 Tähtaja ületanud retseptide annulleerimine

Retseptid kuuluvad automaatselt annulleerimisele, kui ZDR\_BATCH-ZANN != INITIAL.

Viia läbi retsepti annulleerimise toiming vastavalt spetsifikatsioonis ZCORE peatükile 9 (Retsepti annulleerimine) järgmiste PO-de korral

- ZZSTATUS EQ „00” (välja kirjutatud) ning
- EKKO-KDATE < today().

Iga väärtustatud rea kohta kutsuda välja funktsiooni &\_ZLOG& järgmiste parameetrite väärtustega (juhul, kui ZDR\_BATCH-ZLOGACT != INITIAL):

- &TR\_TYPE& ← „40”,
- &TR\_DATE& ← today(),
- &TR\_TIME& ← jooksev kellaaeg,
- &TR\_PARM1& ← EKKO-EBELN,
- &TR\_PARM2& ← EKKO-BEDAT,
- &TR\_VAL1& ← EKKO-KDATB,
- &TR\_VAL2& ← EKKO-KDATE,
- &TR\_VAL3& ← „99” (staatus: annulleeritud).

Omistada ZDR\_BATCH-ZANN\_DATE väärtus today().

## 3 Logifunktsioon

Logifunktsiooni &\_ZLOG& kutsutakse välja järgmiste parameetritega:

- toimingu tüüp &TR\_TYPE&,
- toimingu teostamise kuupäev &TR\_DATE& ja
- kellaaeg &TR\_TIME&,
- toimingu 1.objekt (nt. meditsiiniseadme kaardi või PO number) &TR\_PARM1&,
- toimingu 2.objekt (nt. isikukood) &TR\_PARM2&,
- mingi väärtus 1 (nt. uuendatud rahaline limiit vms) &TR\_VAL1&,
- mingi väärtus 2 (nt. uuendatud kogus vms) &TR\_VAL2&,

- mingi väärtus 3. (nt. omistatud staatus) &TR\_VAL3&.

Logifunktsioon teeb tuima inserti tabelisse ZDT\_BATCH\_LOG kasutades järgmiseid mäppinguid:

- ZTR ← &TR\_TYPE&,
- ZDATS ← &TR\_DATE&,
- ZTIMS ← &TR\_TIME&,
- ZPARAM1 ← &TR\_PARM1&,
- ZPARAM2 ← &TR\_PARM2&,
- ZVAL1 ← &TR\_VAL1&,
- ZVAL2 ← &TR\_VAL2&,
- ZVAL3 ← &TR\_VAL3&.