

# UPRAVLJANJE ORACLE JOBOVIMA POMOĆU ENTERPRISE JOB SCHEDULERA



Rajko Kuzma, dipl. ing.  
rajko.kuzma@intesasanpaolocard.com

## SAŽETAK

U velikim sustavima poput sustava za kartično poslovanje rade se razne obrade podataka kao što su generiranje faktura, uplata i terećenja bankovnih računa, te knjiženja. Nameće se izazov kako upravljati velikim brojem obrada, kako optimalno koristiti Oracle bazu, te kako imati pregled nad svim obradama. Batch procesi (jobovi) koje podržava Oracle nisu bili dovoljni za naše poslovanje, pa smo uveli enterprise job scheduler sustav – EJS. Time smo dinamički povezali obrade na Oracle bazi sa pripadnim ulaznim, tj. izlaznim datotekama i njihovim file transferima prema partnerima. Pojednostavljeno nadgledanje obrada olakšava operativni rad što povećava kvalitetu pruženih usluga.

## ORACLE JOB MANAGEMENT USING ENTERPRISE JOB SCHEDULER

### SUMMARY

In large systems like card business systems, various data processing is going on such as invoice generation, payments and charges to bank accounts. There is a challenge how to manage a large number of data processing, how to optimally use the Oracle database, and how to keep an overview of all processing. Batch processes (jobs), supported by Oracle were not sufficient, so we have introduced enterprise job scheduler - EJS system to our business. This dynamically connects processing at the Oracle database with the corresponding input or output files. Simplified monitoring of data processing makes operational work more efficient, which in turn increases the quality of services provided.

## 1. UVOD

U referatu opisani su načini korištenja Enterprise Job Schedulera kao nadogradnja Oracle-ovog batch mehanizma. Nadalje, osnovni procesi kartičarstva i načini primjene Enterprise Job Scheduling sustava za poboljšanje poslovnog procesa obrada, čiji su temelji na Oracle bazi podataka, kako upravljati većom količinom obrada kao i analiza takvih obrada.

## 2. BATCH PROCESIRANJE – OBRADA PODATAKA

Kartične kuće rade svakodnevno velike obrade podataka. Tako i mi, svakodnevno pokrećemo skup procedura i funkcija koje čine poslovni proces.

Intesa Sanpaolo Card radi procesiranje za više tipova kartica. To su Mastercard, Visa, American Express (Slika 1) te debitne kartice pojedinih banaka.



Slika 1 Primjer plastika kreditnih kartica

Radimo acquiring, issuing te billing. Za svaki produkt koji nudimo na tržištu potrebno je raditi specijalno definirane obrade. Obrada je definirana poslovnim procesom. Takav se proces može podijeliti na niz podprocesa rekurzivno.

Glavni sustav za kartično poslovanje zove se Card Management System koji je izgrađen na Oracle platformi. Ovaj sustav je vrlo kompleksan kao i model poslovnog procesa koji podržava sve te produkte.

Issuer je izdavatelj kartica, banka ili kartična kuća ovlaštena od strane American Express, MasterCard i Visa globalnih kartičnih kuća za izdavanje kartica na lokalnom tržištu. PBZ Card se na hrvatskom tržištu pojavljuje kao kartična kuća Issuer i izdaje široku paletu American Express, MasterCard, Maestro i Visa kartica s raznim pogodnostima prilagođenim za korisnike kartica.

Acquirer je prihvatitelj kartica, banka ili kartična kuća ovlaštena od strane American Express, MasterCard i Visa globalnih kartičnih kuća za sklapanje ugovora o prihvaćanju kartica na prodajnim mjestima. PBZ Card se na hrvatskom tržištu pojavljuje kao kartična kuća - Acquirer i sklapa ugovore za prihvata American Express, MasterCard, Maestro i Visa kartica za fizička i internetska prodajna mjesta.

Billing je spona između issuinga i acquiringa kojim se brine o naplati i potraživanju prodajnih mjesta te svakog nositelja kartice.

## 3. PODATKOVNI KANALI

Postoje više vrste podatkovnih kanala. Njihova glavna podjela se svodi na ulazne i izlazne podatkovne kanale. Oracle baza podataka se puni sa različitih izvora. To su sustavi autorizacija, web

sučelja internet bankarstva te datoteke ostalih partnera. U ulazni podatkovni kanal ulaze i podaci koje generira POS uređaj na prodajnim mjestima kao i bankomati (Slika 2).



**Slika 2** POS i ATM (bankomat) uređaj

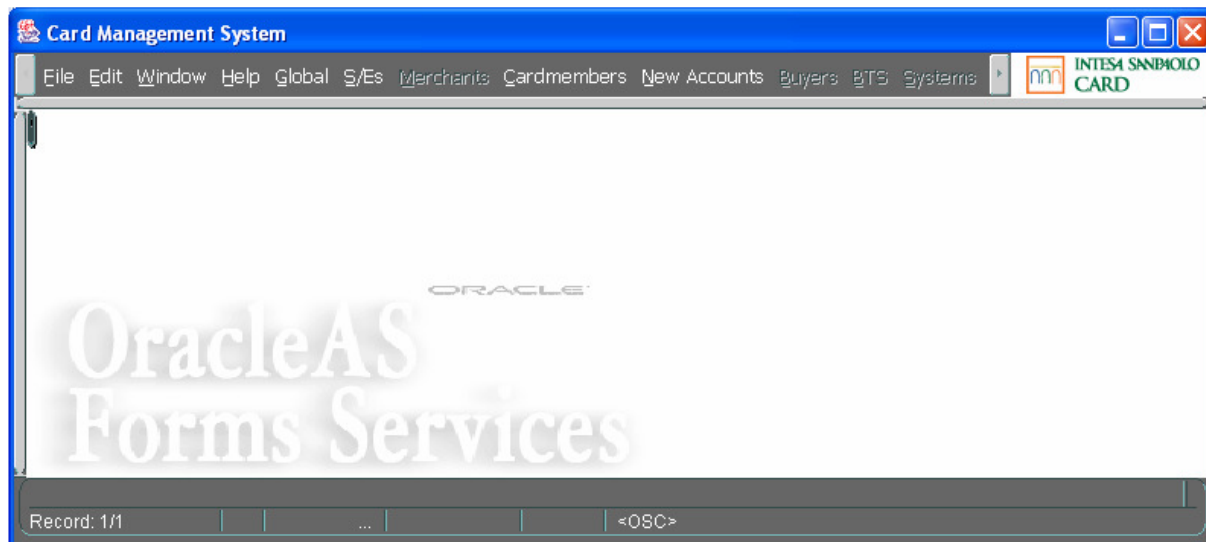
Pošto je velik broj POS uređaja i velika je mreža bankomata, time se generira i velik broj ulaznih datoteka koje je potrebno zapisati u Card Management sustav. Kako smo podijeljeni u timove, tako se taj velik broj podataka prije masovnog punjena Card Management sustava konsolidira od verifikacijskih modula POS odnosno bankomatskog tima.

Bankarstvo, odnosno, kartično poslovanje je prisutno svugdje u svijetu. Kako bi strani klijenti mogli koristiti svoju karticu kod nas, odnosno da naši klijenti mogu koristiti izvan naše mreže, potrebno je imati složenu razmjenu tih podataka. U to spada autorizacijski dio te financijski da se može uredno platiti potraživanje prodajnom mjestu odnosno teretiti pravog člana (nositelja kartice).

#### 4. FRONT END I BACK END CARD MANAGEMENT SUSTAVA

Kako je Card Management sustav napravljen radi, primarnog posla kojeg naša kuća radi, kartičarstva, tako je podijeljen u segmente. Korisnici sustava su naši partneri. Oni se preko Oracle grafičkog sučelja (Oracle forme) prijavljuju na sustav (Slika 3) i na taj način obavljaju svakodnevni operativni posao.

U korisnički posao spada npr. pregled prometa kojeg zahtjeva korisnik kartice, promjena visini iznosa limita kojeg je moguće po pojedinoj kartici napraviti te veliki niz drugih funkcija.



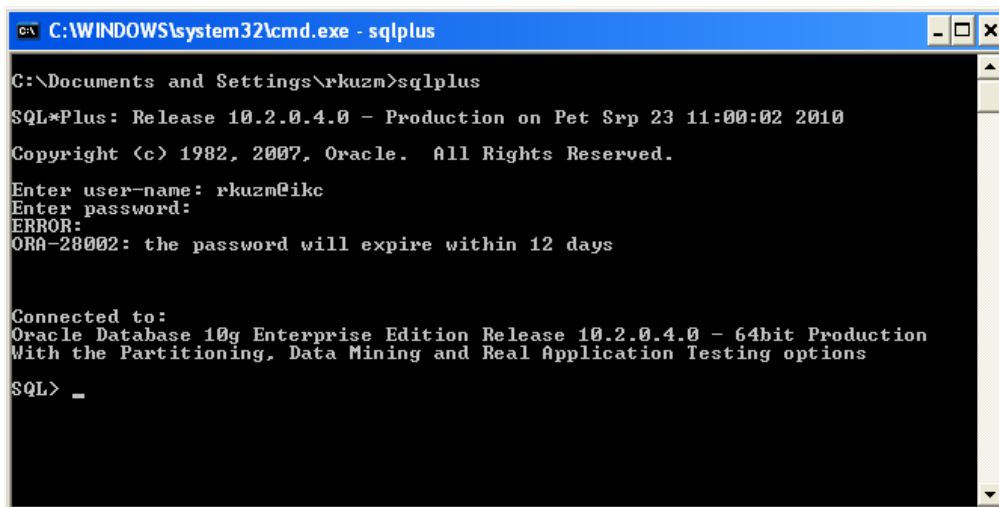
**Slika 3** Oracle grafičko sučelje Card Management sustava

S druge strane, u operativni posao spada i monitoriranje Oracle batch jobova. Ovo je potrebno pratiti zbog toga što postoji mogućnost da se Oracle job (procedura na bazi) ne završi uspješno, već je potrebno ručno intervenirati. Ovaj dio je direktno vezan na podatke koje je potrebno pokupiti sa kanala koji su opisani u prethodnom poglavlju. Monitoriranje odnosno praćenje izvođenja Oracle jobova se radi na puno načina. Preko exception handlera tih procedura, raznih logova te izlaznih datoteka.

Kako smo povećavali broj produkata koje nudimo tržištu tako se ekvivalentno povećava broj batch obrada koje je potrebno pokrenuti na Card Management sustavu.

Optimizacijom takvog poslovnog procesa doveli smo operativni posao u krajnje granice. Pošto je bitno voditi brigu o redoslijedu pokretanja takvih baznih procedura operateri moraju biti oprezni. Na takav način veliki utjecaj na izvođenje batch obrada ima ljudski faktor greške. Batch obrade se rade 24 sata na dan te 7 dana u tjednu i veliki dio ovisi o vanjskim ulaznim podacima. Radi bolje kvalitete operateri su dodatno radili provjere o pristiglim datotekama. Tek tada, kada su svi preduvjeti za pokretanje Oracle bazne procedure zadovoljeni, moguće je pokrenuti Oracle procedure na način da se na bazu spoje sqlplus klijentom (Slika 4). Radili su se manualni planovi izvođenja. Kada se radilo točno po planu, rijetko kad je bilo vremena da se ranije nego dođe do trenutka obrade provjeri da li su svi potrebni ulazni podaci pristigli do nas.

Kako smo mi vezani SLA ugovorima potrebno je bilo da implementiramo viši i moćniji sustav kojime bi upravljali batch jobovima, provjeravali ulazne podatke i datoteke, te sve zajedno nadgledali, tj. monitorirali na najefikasniji način.



```

C:\WINDOWS\system32\cmd.exe - sqlplus
C:\Documents and Settings\rkuzm>sqlplus
SQL*Plus: Release 10.2.0.4.0 - Production on Pet Srp 23 11:00:02 2010
Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
Enter user-name: rkuzm@ikc
Enter password:
ERROR:
ORA-28002: the password will expire within 12 days

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit Production
With the Partitioning, Data Mining and Real Application Testing options
SQL> _
    
```

Slika 4 Konekcija na bazu, sqlplus

Izlaz obrada su generirane fakture, ažurirana stanja računa, te razni popratni izvještaji. Isto tako generiramo izlazne datoteke koje šaljemo našim partnerima, odnosno, VISA-i, Master Card-u te Global Network Services-u.

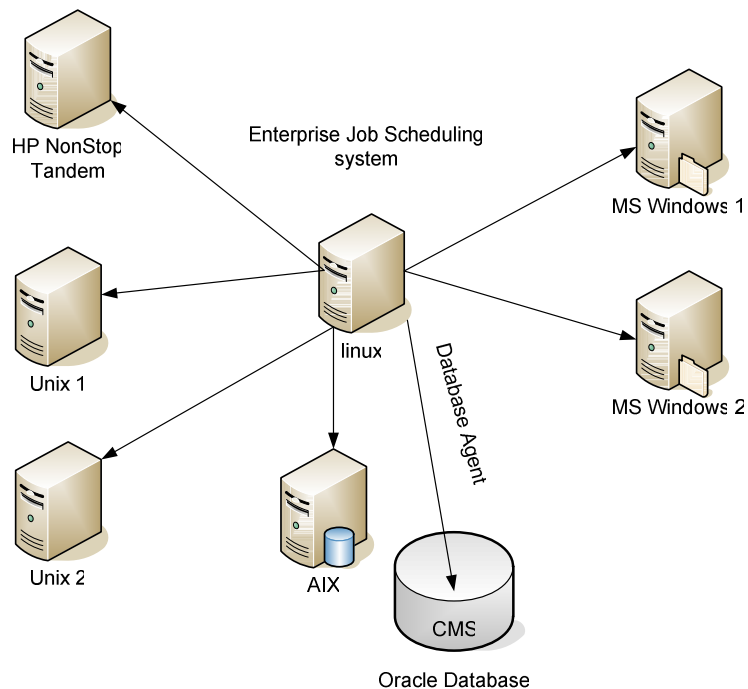
Naši partneri žele imati dodatne izvještaje tih obrada, pa su operateri uz pokretanja Oracle jobova morali i brinuti o kreiranju i slanju izvještaja. Ti su se izvještaji većinom kreirali ručno, te isti slali putem mail-a.

## 5. ENTERPRISE JOB SCHEDULER – UC4

Enterprise Job Scheduling sustav UC4 je dizajniran na način da se sam brine o izvođenju svih vrsta jobova koji se definiraju u njemu. Nije namijenjen da komunicira s operaterom u smislu da mu zadaje parametre ili slične ulazne podatke, ali podržava interakciju u smislu da operateru daje mogućnost upravljanja tokom izvođenja. U slijedećim poglavljima biti će opisane glavne karakteristike ovog sustava.

### 5.1. Arhitektura EJS sustava

Centralni dio sustava je na linux aplikacijskom serveru. Sa ovog mjesta (Slika 5) se inicijaliziraju svi poslovi koje Enterprise Job Scheduling sustav mora odraditi. Za pristup toj aplikaciji potrebna je upravljačka konzola, tj. tzv. *Dialog client*. Sa tim se klijentom radi razvoj poslovnog procesa kojeg EJS sustav mora odrađivati, upravljanje, te monitoring, tj. nadziranje statusa jobova na cijelom sustavu.

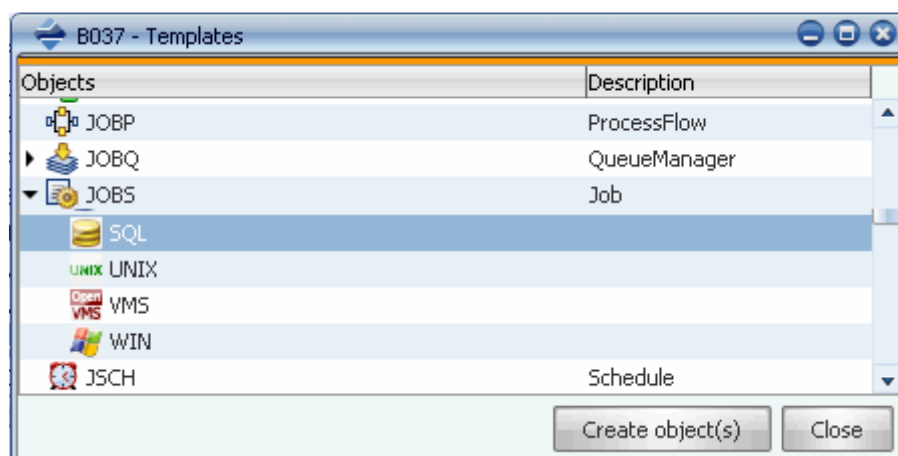


**Slika 5** Arhitektura EJS sustava

Agenti su radni moduli Enterprise Job Scheduling sustava. Oni zapravo rade posao koje im centralni sustav zadaje. Postoje razlike između agenata. Moguće ih je razvrstati na više načina. Kojem operacijskom sustavu pripada (Unix/AIX, HP NonStop - Tandem, MS Windows), kojim bazama može pristupiti (Oracle) te u ostale vrste za specifičnu namjenu npr. reporting.

## 5.2. Vrste Jobova

Enterprise Job Scheduling sustav podržava više tipova jobova (Slika 6). Ovdje će biti riječ samo o nekim najinteresantnijim za spomenuti. Glavna podjela vrste jobova je ovisna o kojoj je platformi riječ (koji je tip agenta), odnosno o kojoj bazi.



**Slika 6** Lista najvažnijih objekata

### Database (Oracle)

- ovim se jobom rade zadaci nad Oracle bazom
- pokretanje Oracle funkcija i Oracle procedura
- u tekstu kasnije će ovaj dio biti detaljnije opisani

#### Unix

- ovim se jobom rade zadaci nad grupom unix operacijskih sustava
- to su shell naredbe na unix, linux, AIX sustavima

#### Windows

- ovim se jobom rade zadaci na MS Windows operacijskom sustavu
- to su razne naredbe iz komande linije

#### File Transfer

Ovo je vrlo zgodan job u svijetu razvoja Card Management sustava. Podržava prijenos datoteka između bilo koja dva agenta u Enterprise Job Scheduling sustavu.

- služi za dohvat ulaznih datoteka
- izlazne datoteke stavlja na servere kojima mogu partneri pristupiti

Za modeliranje i dodatno upravljanje procesom izvođenja služe tipovi navedeni niže.

#### JobPlan (Process Flow)

- služi za logičko povezivanje pojedinih segmenata (jobova) u jedan funkcionalni niz
- moguća višestruka iskoristivost, tj. višestruko pozivanje istog JobPlana
- u tekstu niže biti će još riječi o ovome

#### Script

- script objekt služi za pisanje EJS programskog kôda neovisno o agentu
- na primjer, za računanje parametara koje batch proces treba za izvođenje

#### Vara

- u ovaj se objekt spremaju parametri tokom obrade, odnosno neposredno prije
- ima nalik na globalne objekte i vanjske datoteke na file serveru

### 5.3. Razvojne okoline

Intesa Sanpaolo Card ima vrlo dobro definiran način razvoja Oracle aplikacija. Cijeli programski kôd se verzionira. Proces kreiranja novog programskog kôda ili izmjena postojećeg podržava standard razvoja na način da postoje više vrsta okolina. To je okolina za razvoj gdje se radi razvoj novih funkcionalnosti i razni unit testovi. Slijedeća okolina služi za testiranja programskog kôda te razna funkcionalna testiranja. Tek dok se provjeri ispravnost nove funkcionalnosti, tj. novog kôda, pristupa se slijedećoj okolini – produkcijskoj okolini.

Ovaj način razvoja zadovoljava i Enterprise Job Scheduling sustav. Paralelno razvoju Oracle kôda se razvija kôd za upravljanjem tim Oracle kôdom. Definiran je proces integracije na koji način se prebacuju razni objekti npr. s razvojnog okruženja na testno okruženje. U ovaj sustav integracije spada i verzioniranje programskog kôda EJS sustava.

Agenti između okolina nisu transparentni, pa je potrebno dodatno voditi brigu prilikom dizajniranja poslova kako će EJS sustav znati o kojoj je okolini riječ.

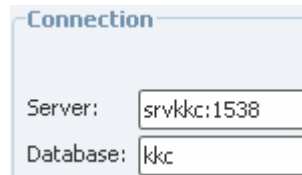
### 5.4. Pozivanje Oracle procedura

Poziv Oracle procedure je trivijalna akcija. Database user se spoji na Oracle bazu preko, na primjer, sqlplus-a te ako ima privilegije za pokretanje procedure lako to i napravi. Isto tako trivijalno je pokrenuti proceduru koja prima, tj. vraća određen broj argumenta.

Kod Enterprise Job Scheduling sustava ova se trivijalna akcija radi skoro na identičan način. Uz definiranje database usera i dodjeljivanje prava, potrebno je definirati parametre (Slika 7) prema Oracle bazi (nalik na tnsnames entry s Oracle clientom) i imati prava za spajanje na database agent.

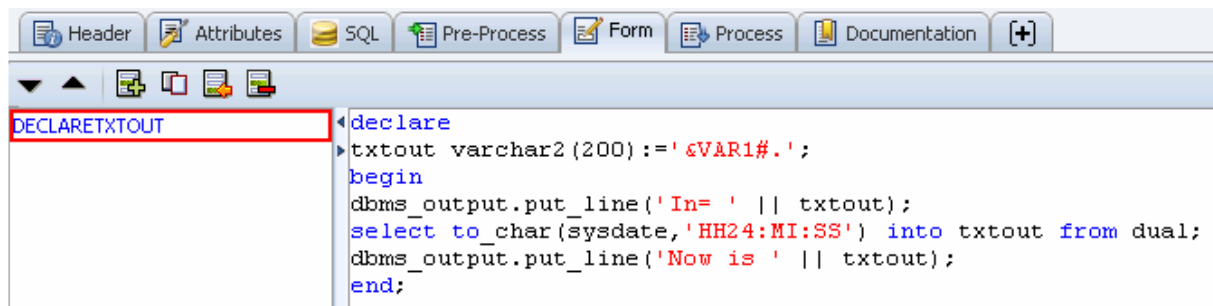
Samo prosljeđivanje parametara u proceduru i dohvat parametara koju ona vraća je ista. Prije svakog poziva potrebno je upisati ulazni parametar te spremiti trenutno stanje. Slično je sa izlaznim parametrima, kad te podatke Enterprise Job Scheduling sustav treba dalje obrađivati, potrebno je dodatno napraviti mehanizam da sustav zna prepoznati koje su to vrijednosti.

Sve ove preduvjete podržava database objekt u Enterprise Job Scheduling sustavu. Poslove koje je potrebno obaviti prije samog poziva (priprema ulaznih parametara) se rade u Pre-process dijelu. Glavni posao, tj. rad nad Oracle bazom potrebno je opisati u Process dijelu. Nakon izvođenja u Post-process dijelu je potrebno napraviti preuzimanje parametara koje Oracle baza vraća te isto provjeriti da li nije Oracle javio neki exception kojeg je potrebno naknadno obraditi.



Slika 7 Definicija konekcije na bazu

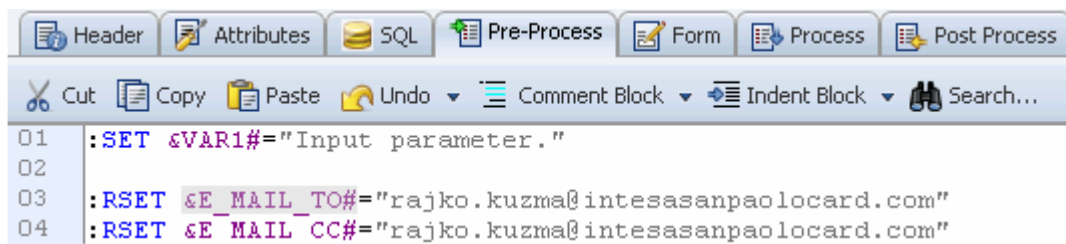
Primjer poziva Oracle procedure – JOBS.ORA:



Slika 8 Poziv Oracle procedure iz EJS sustava

Slika 8 prikazuje na koji način se radi poziv Oracle procedure preko EJS sustava. Pretpostavimo da se ulazni i izlazni parametri ispravno preuzimaju. Uočljivo je da se radi o PL/SQL neimenovanoj proceduri, te kombinacija PL/SQL-a i UC4 programskog kôda koji je interface između Oracle baze i samog Enterprise Job Scheduling sustava (varijabla &VAR1# je ulaz u Oracle proceduru).

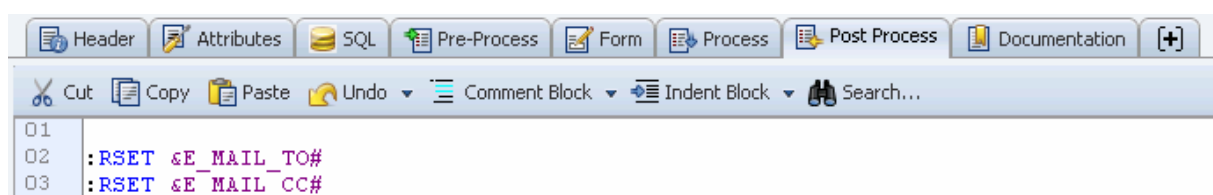
Primjer pripreme ulaznih argumenata:



Slika 9 Priprema ulaznih parametara za poziv Oracle procedure

Za programsko prosljeđivanje parametara u Oracle proceduru koristi se način prikazan na slici Slika 9. Ovdje je moguće radi razne pripreme koje treba prirediti prije nego proces treba Oracle bazu.

Primjer preuzimanja izlaza procedura:



Slika 10 Post Process dio, ručna kontrola

Za programsko preuzimanje parametara iz Oracle procedure kao i provjera ispravnosti izvođenja joba koristi dio objekta prikazana slikom Slika 10, a sam primjer kôda za tu akciju niže. Time se može napraviti automatsko popravljavanje greške, npr. ponoviti slanje ako u prvom koraku je došlo do greške ili forsirano dodatno slati mail podršci da je došlo do neočekivanog problema.

```

:RSET &E_MAIL_TO#
:RSET &E_MAIL_CC#

:SET &return_code=GET_UC_OBJECT_STATUS(, "RETCODE")
:print "return_code=&return_code"
:SET &MY_NAME# = SYS_ACT_ME_NAME()
:SET &RUN# = GET_UC_OBJECT_NR(MM.END.PROCESSING)
!:SET &PARENT_NAME# = GET_PARENT_NAME(&RUN#)
:set &NL#=UC_CRLF()
:SET &RET1#=""

:IF &return_code > 0

:   set &MAIL_MESSAGE#=STR_CAT("&MY_NAME# return code = ", "&return_code")
:   set &MAIL_MESSAGE#=STR_CAT("&MAIL_MESSAGE#", "&NL#")
:   SET &HND1# = PREP_PROCESS_REPORT(, "ACT", " *ORA-*")
:   PROCESS &HND1#
:   IF STR_LENGTH(&MAIL_MESSAGE#)<200
:     IF STR_LENGTH(&RET1#)<150
:       SET &RET1# = GET_PROCESS_LINE(&HND1#)
:       set &MAIL_MESSAGE#=STR_CAT("&MAIL_MESSAGE#", "&RET1#")
:       set &MAIL_MESSAGE#=STR_CAT("&MAIL_MESSAGE#", "&NL#")
:     ENDIF
:   ENDIF
:   ENDPROCESS
:   CLOSE_PROCESS &HND1#
:   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:   !!!! send mail while return code is > 0 !!!!!!!!!!!!!!!
:   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:   set &temp=send_mail("&E_MAIL_TO#", "&E_MAIL_CC#", "UC4 report: %MY_NAME#
Error", "&MAIL_MESSAGE#")

:ELSE

:   set &MAIL_MESSAGE# = "Execution of '&MY_NAME#' is OK."
:   SET &HND1# = PREP_PROCESS_REPORT(, "REP", "Now is*")
:   PROCESS &HND1#
:   SET &RET1# = GET_PROCESS_LINE(&HND1#)
:   IF STR_LENGTH(&MAIL_MESSAGE#)<200
:     IF STR_LENGTH(&RET1#)<150
:       P &RET1#
:       set &RET1# = STR_SUB(&RET1#, 'Now is ', '')
:       pset &MAIL_MESSAGE#=STR_CAT("&MAIL_MESSAGE#&NL#", ">>&RET1#<<")
:     ENDIF
:   ENDIF
:   ENDPROCESS
:   CLOSE_PROCESS &HND1#
:   !:set &temp=send_mail("&E_MAIL_TO#", "&E_MAIL_CC#", "UC4 report: &MY_NAME#
Success", "&MAIL_MESSAGE#")

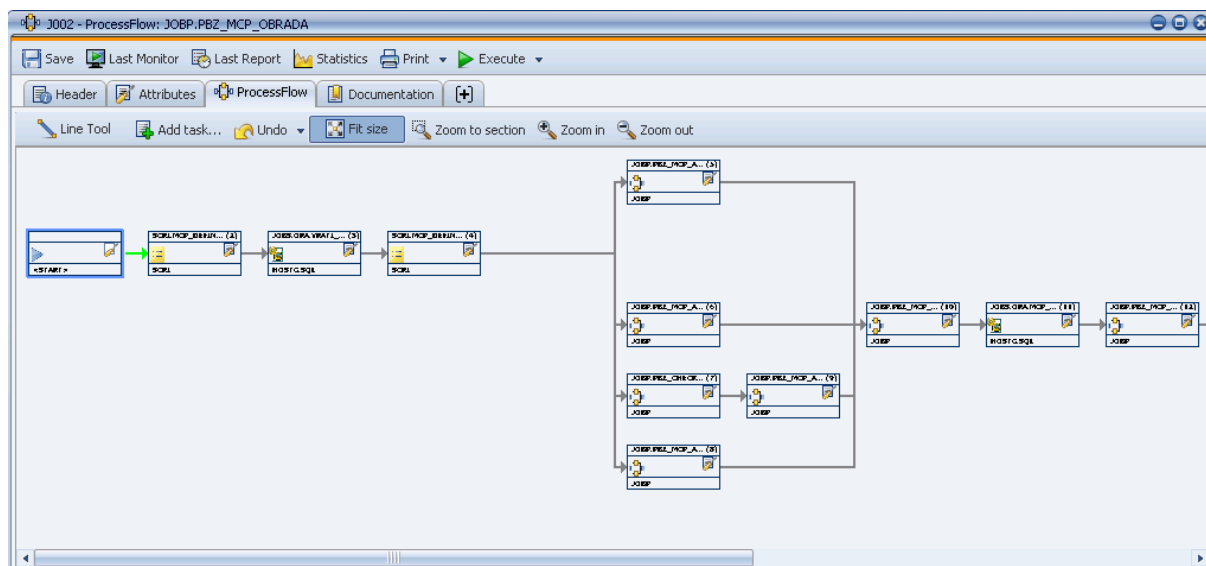
:ENDIF

:print &MAIL_MESSAGE#
    
```



## 5.5. Process Flow – JobPlan

JobPlan je vrsta objekta u Enterprise Job Scheduling sustavu koji služi za logičko povezivanje svih pojedinih jobova (podsegmenta) u funkcionalni skup.



**Slika 11** Process Flow - JobPlan

Slika 11 predstavlja jedan JobPlan, tj. opis segmenta poslovnog procesa. Svaka tzv. „kućica“ predstavlja neki batch job koji se pokreće u točno određenom redoslijedu. To mogu biti Oracle pozivi, pripreme parametara, dohvat datoteka, slanje datoteka, odnosno, file transferi od izvorišta do odredišta. Zgodno je da je moguće cijele JobPlan objekte međusobno povezati. Uočljivo je da se ovime postiže preglednost izvođenja pojedinih procesa te se omogućava da se neki dijelovi mogu paralelno izvesti (ili zabraniti).

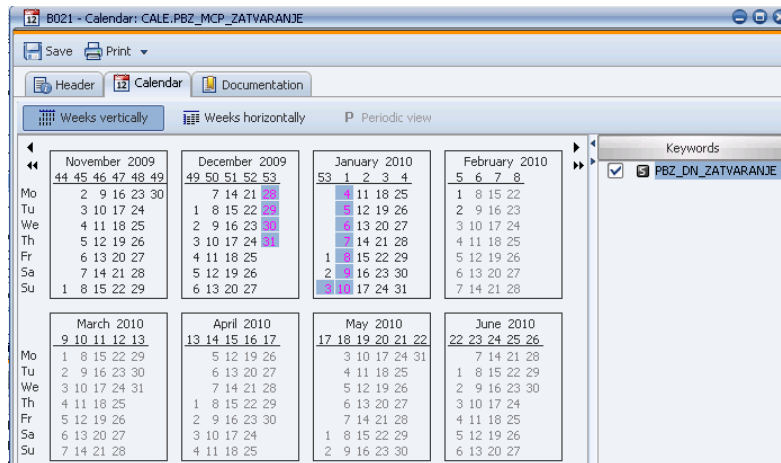
## 5.6. Scheduling mehanizam

Enterprise Job Scheduling sustav ima dobro definirani mehanizam za scheduling. Podržava sve funkcije kao što ih ima Oracle-ov scheduler. Vrlo zanimljiva funkcionalnost je funkcija kalendara. Moguće je definirati kalendar npr. za pojedine države, te dinamički sve jobove koje EJS sustava mora izvršiti čeka da mu kalendar dopusti pokretanje joba. Na primjer, ako se neki jobovi moraju izvršavati prvi radni utorak u mjesecu. Ako se ne želi koristiti mehanizam kalendara, omogućen je rad da se ciklički definira izvođenje jobova ili da se manualnim putem isti pokreću. Praćenje izvođenja će biti opisano u kasnijim poglavljima.

Slika 12 prikazuje svojstva kalendara. Na njemu je definirana jedna statička varijabla koja točno definira dane u vremenskoj domeni. Ti dani su osjenčani na slici.

Uz statičkih indikatora ako što su blagdani i neradni dani, ovdje se mogu definirati i dinamički. To su na primjer dani u tjednu, vikendi i slično.

Dodatni benefiti su da se svi ti indikatora kod definiranja joba mogu kombinirati. Tako se mogu EJS sustavu prepustiti obrade koje se po poslovnom procesu moraju izvesti od ponedjeljka do petka bez da se proces pokrene u tom periodu ako je neradni dan (npr. praznik pada na srijedu).



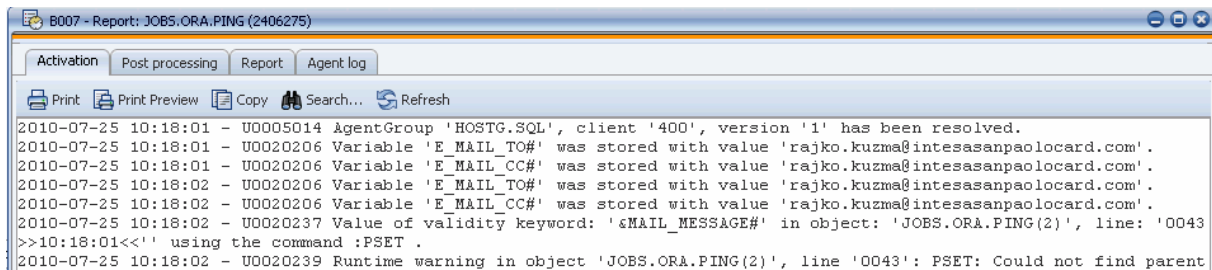
Slika 12 Kalendar, ključna riječ

## 5.7. Nadgledanje procesa izvođenja - monitoring

Kad Enterprise Job Scheduling sustav preuzme kontrolu nad pokretanjem batch jobova, tokom izvođenja generiraju se popratni logovi.

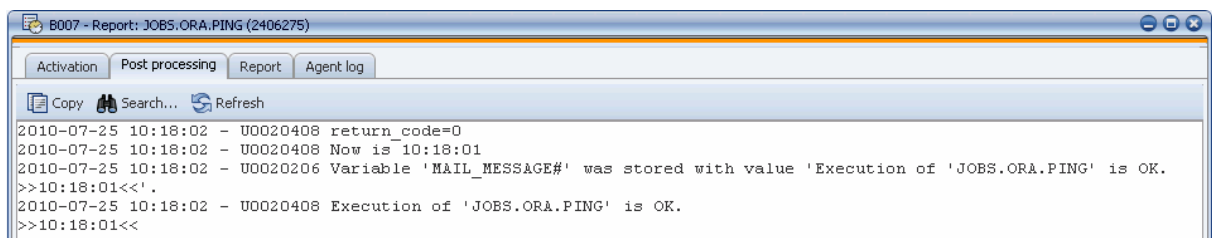
### 5.7.1. Primjer izvještaja

Sustav EJS generira Activation, Post Process, Report i Agent logove. Operater može u bilo kojem trenutku saznati u kojem je stanju pojedini batch proces pregledavanjem tih izvještaja. Slika 13 prikazuje log svake akcije koje EJS sustav napravi, na primjer, pridruživanje vrijednosti varijablama.



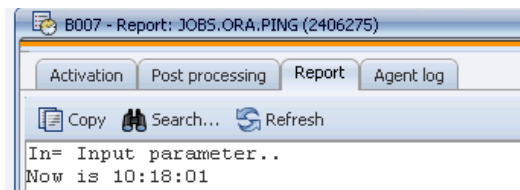
Slika 13 Activation izvještaj

Slika 14 prikazuje Post Process izvještaj. Ako se, na primjer, u Oracle jobu ne koristi Post Process dio, ovaj se izvještaj ne generira.



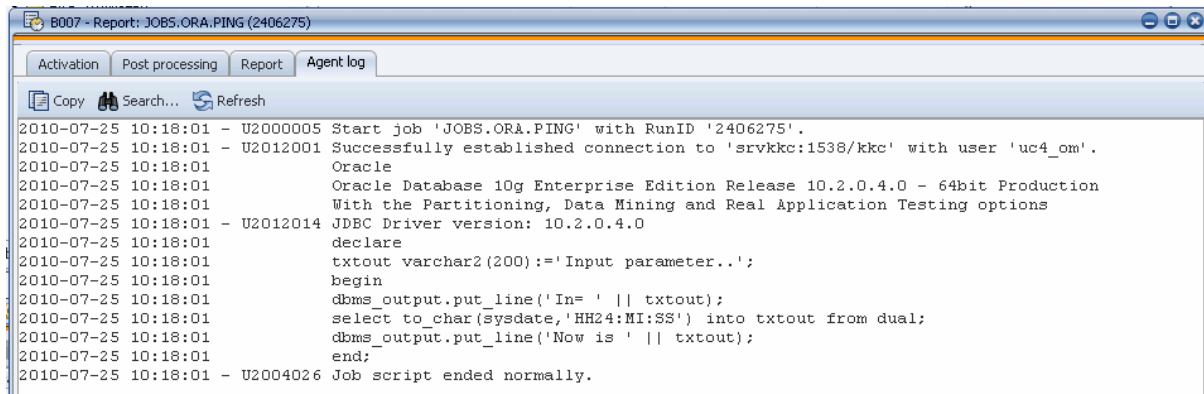
Slika 14 Post Process izvještaj

Sama interakcija s Oracle bazom upisuje se u Report izvještaj. To se može usporediti sa standardnim ulazom kod sqlplus-a. Primjećujemo da su prikazani dbms\_output vrijednosti (Slika 15).

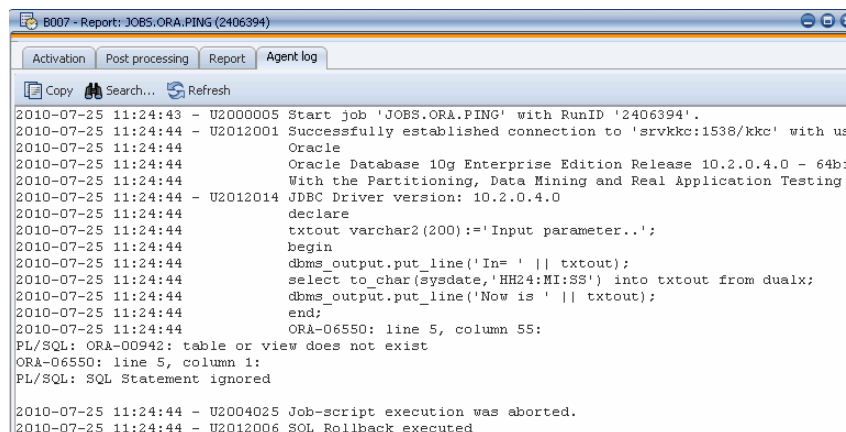


Slika 15 Report izvještaj

Kod database agenta, u izvještaju nazvan Agent log, vidljiva je komunikacija sa Oracle bazom (Slika 16). Ovdje je PL/SQL kôd kojeg Oracle mora izvršiti. Isto se vidi i ORA-xxx exception ako se isti pojavi (Slika 17).



Slika 16 Agent log izvještaj



Slika 17 Agent log izvještaj sa exception-om

## 5.7.2. Activities, Monitor

U Activities listi se nalaze popisani svi jobovi te njihovi trenutni statusi. Za Oracle job može saznati da li se pripremaju ulazni podaci, da li se preuzimaju sa izlaza ili je trenutno glavni dio, odnosno, Oracle procedura u tijeku.

Mogu se napraviti različiti pogledi na Activities listu. Na taj način je moguće promatrati sve Oracle batch procese koji su u tijeku, ili na primjer sve koji čekaju na red izvođenja.

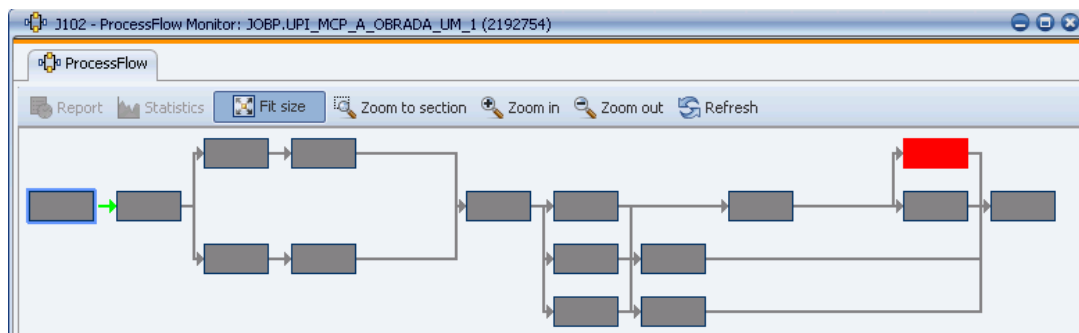
Za svaki pojedini job (stavku u activities listi) pridružuje se i boja. Na taj način pregledom preko tih listi operater može lako provjeriti da li su sve obrade u ispravnom statusu.

Slika 18 prikazuje listu scheduler objekata. Vidljivo je da se uz svaki naziv objekta ispisuju i ID-evi te vrijeme pokretanja kao i trenutni status. Pregledavanje svake stavke operater može saznati više detalja za svaki job koji je zadan Enterprise Job Scheduleru.

Name	Type	RunID	Parent	User	Start time	Prio.	Status
JSCH.PTD	JSCH	2403840		IFRA/PBZCARD	25.07.2010. 00:00:28	200	Active
JSCH.BILLING_CKS_HP22	JSCH	2403841		KKELEMENI/...	25.07.2010. 00:00:28	200	Active
JSCH.BILLING_CKS_HP1	JSCH	2403842		SMARI/PBZC...	25.07.2010. 00:00:28	200	Active
JSCH.BILLING_CKS_SG3	JSCH	2403843		KKELEMENI/...	25.07.2010. 00:00:29	200	Active
JSCH.VUBMPATP	JSCH	2403844		IFRA/PBZCARD	25.07.2010. 00:00:29	200	Active

Slika 18 Activities lista

Svaki proces kojeg EJS sustav preko JobPlan objekta pokreće, moguće je promatrati preko Monitor prozora. Ovim se načinom interaktivno vidi gdje je točno proces i u kojem je statusu. Boje su istovjetne bojana u Activities listi, kao i svaki status joba. Primjećujemo da Slika 19 prikazuje jedan objekt koji je crvene boje. Kad operater uoči tu boju, posvećuje više vremena na to da riješi ovaj slučaj i da se batch proces nastavi izvršavati.

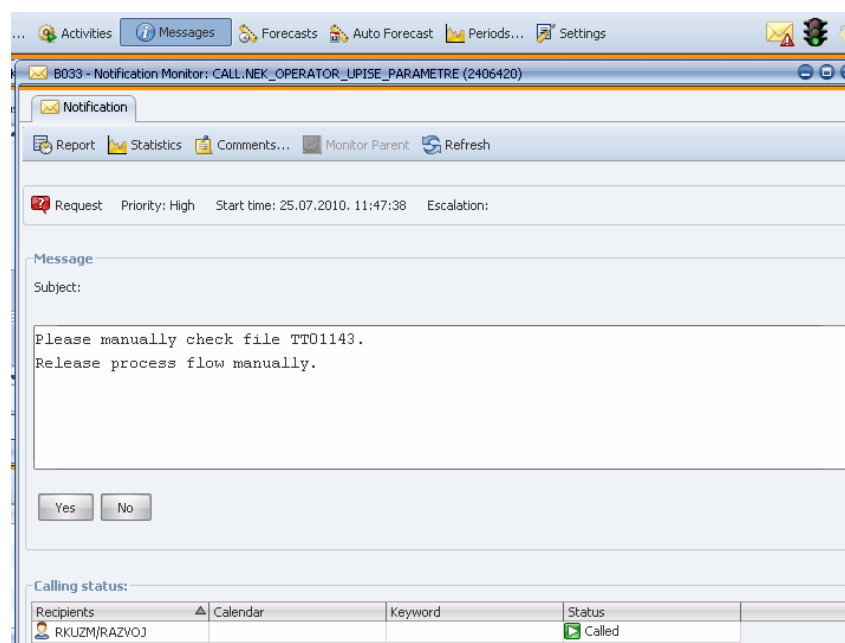


Slika 19 Monitor pogled

### 5.7.3. Call operator mehanizam

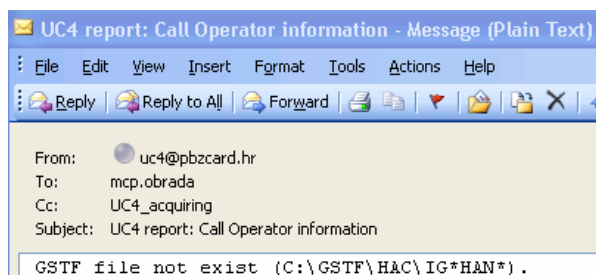
Call operator je prozor kojeg je moguće konfigurirati da javlja operateru kad se desi na primjer neki Oracle exception tokom izvođenja batch procesa. To je interaktivna stanka batch procesiranja i čeka se operatera da nastavi daljnje izvođenje.

Korisno je i kod provjere ulaznih datoteka. Prije pokretanja Oracle joba EJS sustav provjeri da li je ulazna datoteka na ispravnom mjestu, te ako nije, javlja operateru informaciju o nedostatku (Slika 20).



Slika 20 Call operator pop-up

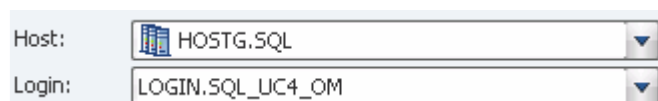
Isto se eskalacije mogu riješiti na ovaj način. Ako se Oracle procedura nekim uzrokom izvršava neuobičajeno dugo, može obavijestiti operatera da provjeri uzrok toga. Popratno pojavljivanju Call operator prozora, mogu se automatski slati e-mail-ovi (Slika 21) na željene grupe korisnika.



**Slika 21** Primjer mail-a od Call operatora

## 5.8. Sigurnost - Security

EJS sustav uc4 je dizajniran na način da podržava točke sigurnosti PCI DSS standarda. Pristup njemu je samo moguće ako administratori sustava dopuste. To vrijedi za korisnike sustava kao i za sam EJS sustav da upravlja batch jobovima.



**Slika 22** Definicija log in parametara

Kod agenta za spajanje na Oracle bazu, radi pokretanja određenih procedura, treba više parametara. To je login informacija na database agent te login parametar na Oracle bazu. Jednom definirani login objekti se mogu koristiti od određene grupe korisnika na način da odabiru njih (Slika 22). Ne moraju znati konjeksijske parametre i passworde, što olakšava korištenje i povećava sigurnost. Prilikom implementacije ovog rješenja, potrebno je dati dovoljna prava za izvođenje database user-u (`GRANT execute`).

### 5.8.1. Administracija

Postoje administracijske konzole za upravljanje agentima, kao i konzola za upravljanje korisnicima. Sustav se veže na LDAP gdje je podržani password policy i izbacuje nepotrebna dodatna administracija korisnika.

U ovu grupu možemo svrstati i upravljanje bazama sa definiranim okolinu u EJS sustavu. Na primjer, razvojno okruženje gleda na razvojnu bazu i analogno tome produkcijsko okruženje EJS sustava na produkcijsku Oracle bazu podataka. Dobrim administriranjem olakšavamo rad sa EJS sustavom, a time i pojednostavljujemo upravljanje poslovnog procesa.

## 6. ZAKLJUČAK

Uvođenjem Enterprise Job Scheduling sustava znatno se olakšava praćenje izvođenja batch jobova. Kako svakodnevno povećavamo broj aktivnih batch jobova, njihovo upravljanje preuzima EJS sustav i on se sam brine o pravom trenutku pokretanja. Faktor ljudske pogreške je sveden na minimum. Riješeno je pravodobno obavještanje operatera u slučaju pogreške te se zastoje na produkcijskoj okolini svodi na minimalno vrijeme. Kad sve ove benefite Enterprise Job Scheduling sustava uzmemo u obzir dolazimo do zaključka da tržištu možemo pružiti vrlo dobre usluge i time bolje konkurirati.