

POSTUPAK FORMIRANJA ELEKTRONSKIH NALOGA ZA PLAĆANJE I TRANSFORMACIJA U XML FORMAT

PROCEDURE FOR ESTABLISHMENT OF ELECTRONIC PAYMENT ORDER AND TRANSFORMATION INTO XML FORMAT

Dušan Trajković, spec.¹
dr Bojan Milosavljević²

Rezime: U ovom radu je, na primeru poslovne aktivnosti obračuna zarada zaposlenih, prikazan postupak formiranja elektronskih naloga iz Clipper baze podataka, **DBF** formata (dBASE), i njihovo smeštanje u fajlove za slanje FlatFile, prema zahtevu Uprave za Trezor. Poseban doprinos ovog rada je prikaz transformacije elektronskih naloga iz **FlatFile** u **XML** (eXtensible Markup Language) format podataka primenom savremenog softverskog alata, Stylus Studio 2010, za XML IDE (Integrated Development Environment). Ovo je od posebnog značaja u postupku slanja elektronskih podataka zbog sve izvesnijeg prelaska obavljanja ovih transakcija putem WEB-a u okviru koga XML postaje standard.

Ključne reči: Uprava za Trezor, Elektronska razmena podataka, Elektronski nalozi, FlatFile, XML, transformacija

Abstract: In this paper, the process of forming electronic orders from a Clipper database in the **DBF** (dBASE) file format and their transformation into FlatFile format for sending, according to the request of the Treasury is shown by example of the business activities for accounting earnings. A special contribution of this paper is to present the transformation of electronic orders from **FlatFile** into **XML** (eXtensible Markup Language) format data using modern software tools, Stylus Studio 2010 XML IDE (Integrated Development Environment). This is of particular importance in the process of sending electronic data due to the potential moving of these transactions through the WEB, in which XML becomes the standard.

Keywords: Department of the Treasury, Electronic Data Interchange, Electronic Orders, FlatFile, XML, transformation

1. UVOD

Sastavni deo poslovanja preduzeća čini plaćanje svojih obaveza prema korisnicima. Prijem, kontrola i obrada naloga za plaćanje direktnih i indirektnih korisnika budžetskih sredstava Republike i lokalnih vlasti, kao i korisnika sredstava organizacija obaveznog socijalnog osiguranja, je u nadležnosti Uprave za trezor [1], kao organa uprave u sastavu Ministarstva finansija Republike Srbije. Ova aktivnost se pored tradicionalnog načina popunjavanja naloga, ručno ili programski odštampano na štampaču, može obavljati i elektronski.

¹ predavač, Visoka škola strukovnih studija za poslovno industrijski menadžment – Kruševac, dusantt@gmail.com

² docent, Fakultet za industrijski menadžment – Kruševac, bmilosavljevic@gmail.com

U slučaju predavanja naloga u papirnoj formi neophodno je ponovno unošenje podataka o izvršenim uplatama na računarima Uprave za trezor. Zbog svakodnevne velike količine naloga za plaćanje javlja se veliki obim posla i samim tim povećava se i mogućnost greške pri ponovnom unosu u računar. Da bi se ovaj problem eliminisao Uprava za trezor je, septembra 2004. godine, definisala strukturu poruke za elektronsko slanje naloga za plaćanje - poruka *NAL100*, koja se koristi kada učesnik u platnom prometu inicira plaćanje.

2. STRUKTURA SLOGA ELEKTRONSKIH NALOGA

Podaci o nalogima za plaćanje bilo da se dostavljaju na disketi ili se šalju elektronski sistemom ISPP moraju biti formirani po sledećim pravilima:

- naziv datoteke sa podacima je: **EXTDATxx** gde je xx broj preseka tj. redni broj datoteke koja se šalje za taj obradni dan;
- brojačani podaci treba da budu zapisani s vodećim nulama i desnim poravnanjem;
- brojčani podaci iznosa zapisani su bez decimalne zapete i zadnja dva znaka predstavljaju decimalna mesta;
- prazni podaci pune se blenkovima;
- alfanumerički podaci imaju levo poravnanje i do dužine polja popunjeni su razmacima (blenkovima).

Elektronski zapis sadrži tri vrste sloga:

- *labelni slog*, koji, pre svega, identifikuje podnosioca elektronske poruke,
- *zbirni slog*, koji, pre svega, prikazuje ukupan iznos i ukupan broj naloga u poruci i
- *individualni slog*, koji detaljno opisuje konkretni nalog za plaćanje.

U tabelama 1, 2 i 3 data je specifikacija za labelni, zbirni i pojedinačni slog respektivno [2].

Tabela 1 - *NAL100* Specifikacija polja za Labelni slog

Rbr	Naziv polja	Tip i dužina polja
1.	Račun podnosioca	N 18
2.	Naziv podnosioca	A 35
3.	Mesto podnosioca	A 10
4.	Datum slanja naloga	DDMMGG, N 6
5.	Slobodno	A 48
6.	Broj preseka - redni broj datoteke poslate u okviru jednog radnog dana	N 2
7.	Ukupno pojedinačnih naloga	N 6
8.	Ukupan iznos (u parama bez decimalnih mesta)	N 18
9.	Slobodno	N 36
10.	Tip sloga (vrednost 0)	N 1
UKUPNO		180

Tabela 2 - *NAL100* Specifikacija polja za Zbirni slog

Rbr	Naziv polja	Tip i dužina polja
1.	Broj računa koji se tereti (može se teretiti samo sopstveni račun)	N 18
2.	Naziv računa	A 35
3.	Mesto računa	A 10
4.	Ukupan iznos	N 15
5.	Ukupan broj naloga	N 5
6.	Slobodno	A 89
7.	Godina	GG, N 2
8.	Redni broj dana u godini	N 3
9.	Vrsta dokumenta - vrednost 13	N 2
10.	Tip sloga (vrednost 9)	N 1
UKUPNO		180

Tabela 3 - NAL100 Specifikacija polja Pojedinačnog sloga

Rbr	Naziv polja	Tip i dužina polja
1.	Broj računa kome se odobravaju sredstva	N 18
2.	Naziv računa	A 35
3.	Mesto računa	A 10
4.	Način izvršenja - ako je nalog hitan oznaka 1	N 1
5.	Broj modela poziva na broj zaduženja	N 2
6.	Poziv na broj zaduženja	A 22
7.	Svrha plaćanja	A 36
8.	Slobodno	A 6
9.	Šifra plaćanja	N 3
10.	Iznos	N 15
11.	Broj modela poziva na broj odobrenja	N 2
12.	Poziv na broj odobrenja	A 22
13.	Datum valute	GGMMDD, N 6
14.	Slobodno	A 1
15.	Tip sloga (vrednost je 1)	N 1
UKUPNO		180

Iza labelnog sloga slede grupe sastavljene od jednog zbirnog i više individualnih slogova. Zbirni slog i pripadajući individualni slogovi čine logičnu celinu podataka za koju važe sledeća pravila:

- u elektronskom zapisu ne može biti više od 999 zbirnih slogova;
- u okviru jednog zbirnog sloga može biti najviše 999 individualnih slogova.

Skupovi znakova koje je moguće koristiti tokom razmene zasnovani su na *ASCII* skupu znakova. Oznaka **N** znači da su znaci cifarski i da su desno poravnati sa vodećim nulama. Oznaka **A** znači da su znaci slova, cifre i dozvoljeni specijalni znaci i da su levo poravnati sa blanko znacima na kraju.

3. FORMIRANJE DATOTEKE ZA SLANJE - FLAT FILE

U ovom radu je dat primer formiranja elektronskih naloga u okviru poslovne aktivnosti obračuna zarada. Poslovna aplikacija je rađena u Clipper programskom jeziku koji koristi **DBF** (*dBASE*) format baze podataka. Nakon izvršenog obračuna zarada zaposlenih, za konkretnu isplatu (akontacija ili konačni deo) za dati mesec, mogu se generisati tri grupe naloga za plaćanje:

- nalozi za uplatu poreza i doprinosa;
- nalozi obustava iz zarade po osnovu kredita i samodoprinosu;
- nalozi za uplatu neto zarada zaposlenih po tekućim računima.

U zavisnosti od preduzeća ovaj broj naloga u okviru jedne isplate može da se kreće od par desetina naloga pa čak i preko 200 naloga po isplati. Broj naloga najviše zavisi od obima obustava, a može da se i udvostruči, u odnosu na napred pomenuti broj, u slučaju isplate po dva računa (iz sopstvenih sredstava i iz budžetskih sredstava) kada se svaki nalog duplira zbog dva računa.

U postupku formiranja elektronskih naloga najpre se vrši izdvajanje potrebnih podataka iz **DBF** baze i njihovo smeštanje u slogove fiksne dužine, kako je već definisano zadatom strukturom slogova. Programska procedura koja vrši formiranje elektronskih naloga prikazana je na slici 1.

Tako spremljeni slogovi se čuvaju u tekstualnom fajlu pod nazivom **EXTDAT01.TXT**, koji, s obzirom na njegove karakteristike, predstavlja **FlatFile** datoteku, *datoteku izdvojenih slogova fiksne dužine u tekstualnom formatu* [3]. Na slici 2 dat je prikaz sadržaja dela FlatFile datoteke spremne za slanje Upravi za trezor.

```

static procedure enalform()
local racpod, br:=0, rbrdat, fajl, datnow, datenal, brnal, izndat

xpdel('extdat*.txt')

select nal100; set order to 2
go top

datnow:=dtos(date())
datenal=substr(datnow,7,2)+substr(datnow,5,2)+substr(datnow,3,2)

set console off
while leof()
rbrdat:=padl(trim(str(++br,2)),2,"0")
fajl="extdat"+rbrdat //+".txt"
set alternate to &fajl
set alternate on

racpod=racunpod
brnal:=0; izndat:=0

while racunpod==racpod
++brnal
izndat:=izndat+iznos
skip
enddo
seek racpod

?? racunpod+nazivpod+mestopod+datenal+space(48)+rbrdat+padl(brnal,6,"0")
?? stuff(padl(trim(str(izndat,15,2)),19,"0"),17,1,"")*replicate("0",36)+0"

? racunpod+nazivpod+mestopod+stuff(padl(trim(str(izndat,15,2)),16,"0"),14,1,"")
?? padl(brnal,5,"0")+space(89)+substr(datnow,3,2)+padl(trim(str(xpdoy(date)),3),3,"0")+
"13"+"9"

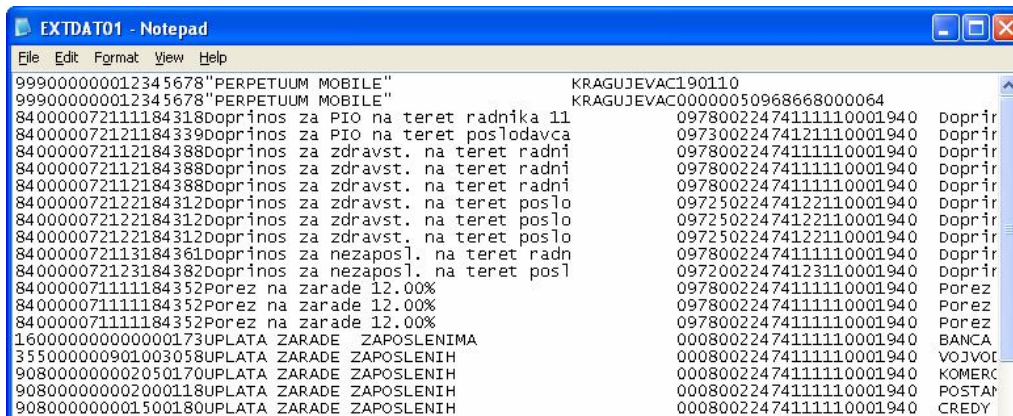
while racunpod==racpod
? racunpla+nazivpla+mestopla+"0"+modpozpad+pozbrojzad+svrhapla
?? space(6)+sifplacnja+stuff(padl(trim(str(iznos,15,2)),16,"0"),14,1,"")
?? modpozod+pozbrojod+datenal+" "+"1"
skip
enddo

set alternate off
set alternate to
enddo
set console on
set order to 1

Xpalert('Poruka', 'Baza elektronskih naloga je uspesno formatirana ;
i spremna je za slanje UPRAVI ZA TREZOR.')
return

```

Slika 1 - Programaska procedura koja vrši formiranje elektronskih naloga u FlatFile [4]



Slika 2 - FlatFile sa formatizovanim eNalozima spremnim za slanje

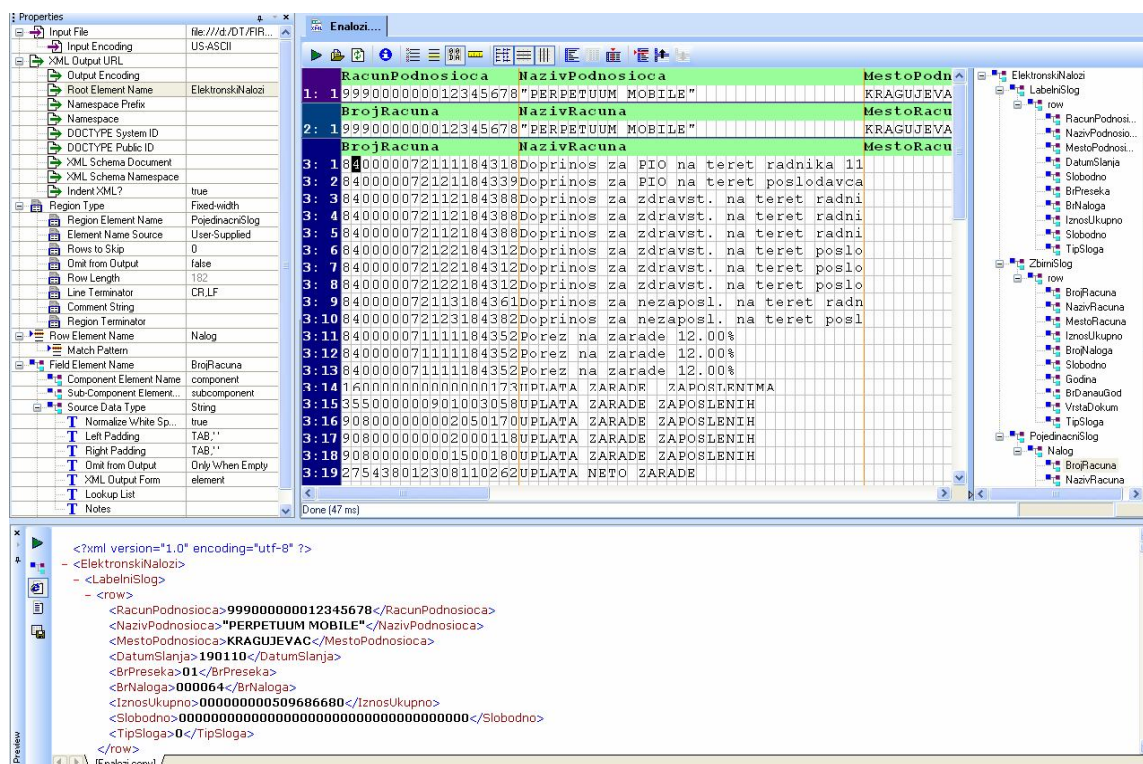
4. TRANSFORMACIJA NALOGA U XML FORMAT

XML je markerski jezik napravljen specijalno za prenos informacija preko WEB-a. Kao skup srodnih tehnologija za moderno poslovanje putem Interneta, posebno u segmentu B2B (Business to Business) poslovanja, XML omogućava skladištenje podataka, njihovu obradu i prikaz na WEB-u. Mogućnost da se podaci odvoje od procesa predstavlja ključ uspeha XML-a [5].

Ono što XML dokument čini posebno upotrebljivim jesu sledeće činjenice:

- u pitanju je javni (nevlasnički) standard za razmenu podataka što znači da se može slobodno koristiti;

- XML je podržan od strane velikih proizvođača softvera čime je i postao široko dostupan i primenjiv;
- u pitanju je otvoreni standard, koji omogućuje da XML funkcioniše na bilo kojoj platformi sa bilo kojim programskim jezikom;
- XML opisuje podatke u tekstualnom formatu te omogućuje razmenu podataka nezavisno od sistema i formata podataka;
- struktura XML dokumenta je takva da pored informacija opisuje i hijerarhijsku strukturu informacija što čini XML upotrebljivim u smislu baze podataka.



Slika 3 - Integrirano razvojno okruženje programa Stylus Studio 2010 [6]

Napred pobrojane pogodnosti XML-a bliže objašnjavaju razlog prevođenja već pripremljene FlatFile datoteke u XML format. U ovom radu se za transformaciju koristi integrirano razvojno okruženje za XML, Stylus Studio 2010 [7] (slika 3).

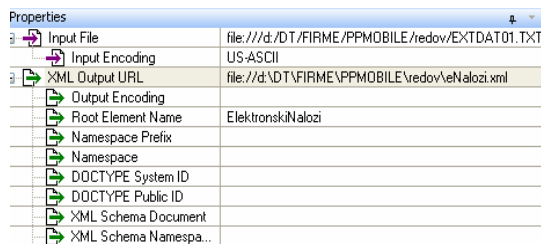
Nakon učitavanja tekstualnog fajla sa elektronskim nalozima, EXTDAT01.TXT, vrši se struktuiranje sadržaja fajla na labelni, zbirni i pojedinačni slog, a potom i samih slogova na polja prema definisanim dužinama. Svakom polju se dodeljuje odgovarajući naziv. Program automatski formira tagove prema imenima polja i automatski formira hijerarhijsku strukturu elemenata (naloga).

Na slici 3 je prikazano razvojno okruženje sa svim potrebnim panelima:

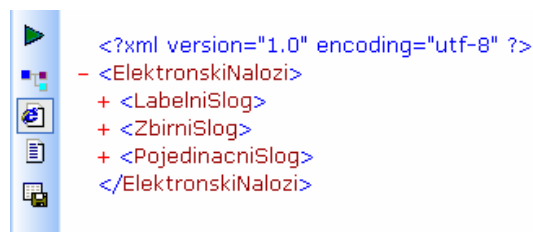
- centralni panel prikazuje sadržaj fajla za transformaciju sa podacima učitanoog TXT fajla;
- levi panel (Properties) prikazuje podešavanja vezana za transformaciju;
- desni panel prikazuje XML hijerarhijska struktura elemenata u obliku stabla;
- donji panel prikazuje konačni sadržaj XML dokumenta sa konkretnim nalozima.

Ovako definisana transformacija se čuva u fajlu tipa CONV (conversion) eNalozi.conv i svakim njenim pokretanjem se sadržaj FlatFile datoteke EXTDAT01.TXT automatski prevodi u XML format čiji se sadržaj snima u fajl pod nazivom eNalozi.xml (XML Output), što se može videti na Properties panelu sa slike 4.

Sadržaj osnovne strukture formiranog XML dokumenta prema vrstama slogova prikazan je na slici 5, a detaljan sadržaj sa konkretnim vrednostima elemenata prikazan je na slici 6.



Slika 4 - Properties panel



Slika 5 - Sadržaj XML dokumenta na nivou slogova



Slika 6 - Detaljan sadržaj XML dokumenta

5. ZAKLJUČAK

Za potrebe elektronske razmene podataka u čestoj su upotrebi tekstualni formati datoteka. I FlatFile i XML format spadaju u tekstualne datoteke strukturisanog sadržaja. FlatFile, korišćen u ovom radu, u sebi sadrži samo sirove podatke o nalogima za plaćanje, strukturisane prema definisanoj specifikaciji Uprave za trezor, bez ikakvog opisa. Da bi se ovi podaci mogli pravilno očitati i upotrebiti na ciljnom mestu potrebno je poznavanje strukture sloga za šta se obično koriste programske procedure za automatsko očitavanje.

Za razliku od FlatFile datoteka, XML dokument sam sebe opisuje, tj. pored podataka o elektronskim nalogima u sebi sadrži i njihov opis, tj. identifikaciju elemenata, kao i hijerarhijsku strukturu elemenata, što omogućava da se struktura XML dokumenta može prikazati u obliku stabla.

Sama upotreba elektronskih naloga, zbog njihovog svakodnevnog obima, je od velikog značaja u sistemu plaćanja, a upotreba XML formata, na efektan i efikasan način uvodi standardizaciju u ovu oblast.

Upotreba savremenih softverskih rešenja, kakav je prikazani XML IDE Stylus Studio, olakšava prevođenje podataka u XML format i može da omogući njihovo integrisanje sa ERP sistemom korisnika, u ovom slučaju Uprave za trezor.

LITERATURA

- [1] <http://www.trezor.gov.rs/>
- [2] Poruke za elektronsku razmenu podataka između uprave i korisnika - Uputstvo, Republika Srbija - Ministarstvo Finansija - Uprava za Trezor, Centrala Beograd, 2004.

- [3] Nikolić, Z., „Elektronska razmena podataka“, ICIM+, Kruševac, 2007.
- [4] Trajković, D., Clipper Poslovna Aplikacija - Obračun Zarada, 2008.
- [5] Young J., M., „Korak po korak XML“, CET Computer Equipment and Trade, Beograd, 2001.
- [6] Stylus Studio 2010 XML Enterprise Suite, 2009.
- [7] <http://www.stylusstudio.com>