

Zakład Usług Informatycznych „OTAGO”

Specyfikacja automatycznego wczytywania i księgowania wyciągów  
dziennych pobieranych z systemu bankowego

wersja 1.3



autor: Tomasz Rosochacki

Gdańsk, 2012-01-23

## Spis treści

1.Wprowadzenie.....	3
2.Sposób wypełniania komunikatu płatniczego. ....	3
2.1.Wirtualny Numer Rachunku.....	3
2.1.1.Przykładowa konstrukcja Wirtualnego Numeru Rachunku.....	3
2.1.2.Wewnętrzna struktura Rozszerzenia Numeru Rachunku. ....	3
3.Pola komunikatu płatniczego, jakie Bank zobowiązuje się przetwarzać i raportować. ....	3
4.Tryb i forma dostarczania do Klienta danych (plików wynikowych) otrzymywanych w wyniku przetwarzania dokumentów płatniczych .....	4
4.1.Dostarczenie pliku. ....	4
4.2.Format pliku. ....	4
4.2.1.Nazwa pliku (propozycja, nazwa powinna być czytelna dla operatora wczytującego plik i gwarantująca jego unikalność). ....	4
4.2.2.Struktura nagłówka w pliku wynikowym. ....	4
4.2.3.Struktura rekordu płatności w pliku wynikowym.....	5
4.2.4.Opis algorytmu naliczania cyfry kontrolnej w nagłówku pliku.....	5
4.2.4.1.Opis algorytmu funkcji wykonywanej na każdej linii pliku. ....	5
4.2.4.2.Wzór implementacji algorytmu w postaci kodu języka C++.....	6

## **1. Wprowadzenie.**

Niniejszy załącznik określa konstrukcję Wirtualnego Numeru Rachunku, pola dokumentu płatniczego, jakie Bank zobowiązuje się przetwarzać i raportować oraz tryb i formę dostarczania do Klienta danych (plików wynikowych) otrzymywanych w wyniku przetwarzania dokumentów płatniczych.

## **2. Sposób wypełniania komunikatu płatniczego.**

### **2.1. Wirtualny Numer Rachunku.**

#### **2.1.1. Przykładowa konstrukcja Wirtualnego Numeru Rachunku.**

Dane wejściowe:

- **10600034** – Numer Rozliczeniowy płatności masowych
- **9841** – czterocyfrowy Identyfikator Rachunku dedykowany dla UM
- **000123456789** – Identyfikator płatnika

Dane dodane:

- **28** – suma kontrolna standardu IBAN wyliczona według załączonego algorytmu

#### **2.1.2. Wewnętrzna struktura Rozszerzenia Numeru Rachunku.**

- 2 znaki – kod podsystemu
- 2 znaki – kod świadczenia
- 8 znaków – numer nadrzędnej karty kontowej

Docelowy wygląd Numeru Rachunku według powyższej konstrukcji:

**28 10600034 9841 000123456789**

## **3. Pola komunikatu płatniczego, jakie Bank zobowiązuje się przetwarzać i raportować.**

Bank zobowiązuje się przetwarzać i raportować zawartość następujących rozpoznawanych pól, które zawierają dane niezbędne dla Klienta do identyfikacji konkretnej Wpłaty Masowej:

- Liczba kontrolna według standardu IBAN
- Numer Rozliczeniowy
- Identyfikator Rachunku
- Rozszerzenie Numeru Rachunku zawierające Identyfikator Płatnika
- Data obciążenia nadawcy
- Data uznania na rachunku
- Kwota transakcji
- Pole Tytułem
- Nazwa zleceniodawcy

## **4. Tryb i forma dostarczania do Klienta danych (plików wynikowych) otrzymywanych w wyniku przetwarzania dokumentów płatniczych**

### **4.1. Dostarczanie pliku.**

Klient otrzymywał będzie pliki wynikowe zawierające szczegóły wszystkich transakcji przeprowadzonych w danym dniu roboczym. Bank zobowiązuje się dostarczyć je Klientowi najpóźniej następnego dnia roboczego do godz. 12:00. **Klient będzie otrzymywał pliki wynikowe dla poszczególnych Identyfikatorów Rachunku.**

### **4.2. Format pliku.**

#### **4.2.1. Nazwa pliku (propozycja, nazwa powinna być czytelna dla operatora wczytującego plik i gwarantująca jego unikalność).**

**umgymmdd.xxx**

Gdzie:

- **umg** – stała określająca Klienta,
- **yymmdd** – data wyciągu bankowego (y – rok, mm – miesiąc, dd – dzień),
- **xxx** –
  - **przykładowo 841, 842** - trzy ostatnie znaki z Identyfikatora Rachunku
  - - przyjmuje stałą wartość, dla pliku zbiorczego

#### 4.2.2. Struktura nagłówka w pliku wynikowym.

Lp	Nazwa Pola	Typ danych	Wielkość	Wymagalność	Opis	Przykładowe dane
1.	Stała identyfikująca pole jako nagłówek	Numeryczne	2	TAK	Stała równa 01 dla nagłówka	01
2.	Identyfikator wyciągu	Numeryczne	20	TAK	Unikalny identyfikator wyciągu w ramach systemu. Może to być np. oznaczenie w formacie IIIIIYYMMDDX, gdzie: <ul style="list-style-type: none"> <li>• IIII – Identyfikator Rachunku</li> <li>• YYYMMDD – data wyciągu bankowego</li> <li>• X – numer pliku dla danego dnia wyciągu (1 – gdy jeden zrzut dziennie, kolejne numery dla kolejnych sesji)</li> </ul>	9841200409221
3.	Data wygenerowania pliku	Data	8	TAK	Data utworzenia pliku w formacie YYYMMDD	20070522
4.	Data wyciągu	Data	8	TAK	Data odpowiadającego plikowi wyciągu bankowego w formacie YYYMMDD	20070522
5.	Konto uznawane	Tekstowe	35	TAK	Konto uznawane łączną kwotą transakcji zawartych w pliku	”06106000760000401290000392”
6.	Nazwa odbiorcy	Tekstowe	150	TAK	Nazwa odbiorcy transakcji	”UM Zielona Gora”
7.	Liczba transakcji	Numeryczne	10	TAK	Liczba wszystkich transakcji zawartych w pliku	177
8.	Kwota łączna transakcji	Numeryczne	16	TAK	Łączna kwota wszystkich transakcji zawartych w pliku w groszach	7864232
9.	Saldo początkowe	Numeryczne	16	NIE	Saldo początkowe z wyciągu bankowego z danego dnia, którego dotyczą transakcje	0
10.	Saldo końcowe	Numeryczne	16	NIE	Saldo końcowe z wyciągu bankowego z danego dnia, którego dotyczą transakcje	7864232
11.	Kwota wpływów	Numeryczne	16	NIE	Suma wszystkich wpływów na rachunku	7864232
12.	Kwota wydatków	Numeryczne	16	NIE	Stała wszystkich wydatków z rachunku	0
13.	Suma kontrolna	Numeryczne	10	NIE	Suma kontrolna wyliczona na podstawie całego pliku (nagłówki + rekordy), generowana według algorytmu z punktu 3.2.4	32554

Tabela 1

#### 4.2.3. Struktura rekordu płatności w pliku wynikowym.

Lp	Nazwa Pola	Typ danych	Wielkość	Wymagalność	Opis	Przykładowe dane
1.	Stała identyfikująca pole jako rekord płatności	Numeryczne	2	TAK	Stała równa 02 dla danych	02
2.	Stała opisująca typ transakcji	Tekstowe	150	TAK	Opis transakcji, Incoming Payment	”Incoming payment”
3.	Identyfikator wyciągu	Numeryczne	20	TAK	Identyfikator wyciągu – pozycja 2 z nagłówka	9841200409221
4.	Numer rachunku	Tekstowe	35	TAK	Wirtualny Numer Rachunku	”28106000349841000123456789”
5.	Unikalny numer transakcji	Numeryczne	30	TAK	Unikalny numer transakcji (dopuszczalne oznaczenie tekstowe)	0434311231000294
6.	Kwota transakcji	Numeryczne	16	TAK	Kwota transakcji w groszach	74782
7.	Data transakcji (zaksięgowania)	Data	8	TAK	Data zaksięgowania operacji w formacie YYYMMDD, ta data powinna być równa = dacie wyciągu (pozycja 4 z nagłówka)	20040922
8.	Data wpływu (uznania)	Data	8	TAK	Data rzeczywistej wpłaty – data złożenia zlecenia w formacie YYYMMDD	20040921
9.	Nazwisko imię/ Nazwa	Tekstowe	150	TAK	Pole „Zleceniodawca”	”Kowalski Jan, ul Kowalska 23, Zielona Gora”
10.	Tytuł zlecenia	Tekstowe	150	TAK	Pole „Tytułem”	”Podatek od Nieruchomości ”

Tabela 2

- Każda linia musi zawierać wymaganą liczbę pól (liczone są separatory), pola niewymagalne mogą być puste, ale muszą być rozdzielające separatory
- Identyfikator wyciągu musi być unikalny w ramach całego systemu (a nie tylko roku czy rachunku)
- Sposób kodowania pliku: ASCII
- Każda linia zakończona: <CR>, <LF> (ASCII = 13, 10)
- Pola w rekordach oddzielone są przecinkiem (ASCII = 44)
- Pola tekstowe zawarte są cudzysłowach (ASCII=34).
- Format daty: YYYYMMDD
- Format kwot: w groszach (np.15000), kwoty powinny mieć wartość dodatnią, wyjątkiem są zwroty z płatności kartą, które powinny mieć wartość ujemną tj. kwota w groszach poprzedzona znakiem minus (ASCII=45).
- Standard kodowania Polskich znaków: MS Windows Code:1250

#### **4.2.4. Opis algorytmu naliczania cyfry kontrolnej w nagłówku pliku.**

Suma kontrolna zawarta w nagłówku jest sumą wyników funkcji wykonanych na każdym rekordzie pliku wynikowego.

##### **4.2.4.1. Opis algorytmu funkcji wykonywanej na każdej linii pliku.**

- Dla każdej linii pliku czytamy kolejno po dwa znaki.
- W przypadku, gdy ilość znaków w linii jest cyfrą nieparzystą a co za tym idzie ostatnia para składa się tylko z jednego znaku, jako drogi znak przyjmujemy znak o wartości ASCII 127.
- Dla każdej pary znaków wykonuje się poniższe działanie:
  - Dla każdego znaku z pary sprawdzana jest jego wartość ASCII.
  - Jeżeli wartość ASCII jest większa niż 127, jako wartość tego znaku przyjmuje się jako 127.
  - Następnie na tak przygotowanych wartościach wykonuje się działanie  $217 * Z1 + Z2$ , gdzie Z to kolejne znaki pary.
- Wynik powyższego działania przeprowadzonego na każdej parze znaku dodaje się do ogólnej sumy tych działań wykonanych na poprzednich parach znaków.
- Po każdym takim działaniu sprawdzane jest czy całkowita suma jest większa od 65434. Jeżeli tak to suma = suma – 65434.
- Suma mnożona jest przez dwa i następnie sprawdzana czy nie przekroczyła liczby 65434. Jeżeli tak to suma = suma – 65434.

##### **4.2.4.2. Wzór implementacji algorytmu w postaci kodu języka C++.**

```
long genSumy(char *info, short dl) /* obliczenie sumy
kontrolnej linii zawierającej dl znaków, info-wskaźnik na pierwszy znak
linii */
{
  long sumak=0, w1=0;

  while (dl > 0) {
    w1 = czytaj(info, dl);
    dl -= 2;
    if ( dl > 0 ) info=info+2;
    sumak += w1;
    if (sumak > 65434)
```

```
    sumak -= 65434;
    sumak *= 2;
    if ( sumak > 65434)
        sumak -= 65434;
    }
    return sumak;
}

long czytaj(char *info, short dl)    /* obliczenie sumy 2 kolejnych
znaków */
{
    long a=0, b=0;

    a = asc(*info);                /* funkcja zwraca kod
    ascii znaku */
    if (dl > 1) b = asc(*(info+1));
    if ( a>127 ) a=127;
    if ( b>127 ) b=127;

    return 217*a+b;
}
```