

Import Specjalny w programie Finanse i Księgowość

1	Wstęp	7
1.1	INFORMACJE OGÓLNE.....	7
1.2	FORMATY DANYCH	7
1.2.1	STRUKTURA DANYCH.....	7
1.2.2	KODOWANIE ZNAKÓW	8
1.2.3	POLA DATY	8
1.2.4	POLA KWOTY	8
1.2.5	POLA ZŁOŻONE	8
1.2.6	POLA SKŁADANE	9
1.3	POJĘCIA PODSTAWOWE.....	9
1.4	SEKCJE WYJŚCIOWE	11
1.4.1	INFORMACJE OGÓLNE.....	11
1.4.2	POLA DANYCH.....	11
1.4.3	POLE \$KLUCZ	12
1.4.4	DOZWOLONE KOMBINACJE SEKCJI I PODSEKCJI	13
1.5	PRZEBIEG PROCESU PRZETWARZANIA DANYCH	15
1.6	ZASADY TWORZENIA SEKCJI WYJŚCIOWYCH	15
1.6.1	KARTOTEKI	15
1.6.2	SEKCJA DOKUMENT	17
1.6.3	SEKCJA ZAPIS.....	18
1.6.4	SEKCJA REJESTR.....	24
1.6.5	SEKCJA POZYCJA VAT.....	26
1.6.6	SEKCJA TRANSAKCJA.....	32
1.6.7	SEKCJA ROZLICZENIE.....	34
1.6.8	SEKCJA ATRYBUTWALUTA.....	35
1.6.9	ZASADY POWIĄZANIA ROZRACHUNKÓW Z ZAPISAMI	36
1.7	WARIANTY SEKCJI WYJŚCIOWEJ DOKUMENT WRAZ Z JEJ PODSEKCJAMI.	38
1.8	SPOSÓB BUDOWANIA XSLT DLA SZABLONÓW DLA PROFILI EDS.....	38
1.9	WYKORZYSTANIE XPATH W XSLT.....	38
2	Uzgodnienia.....	39
2.1	UZGODNIENIA IMPORTU Z EDS	39
3	Informacje zwrotne.....	40
3.1	ROZSZERZENIE OD WERSJI 2018.2.....	42

4	Format danych wejściowych	42
4.1	DEFINIOVANIE FORMATU PLIKÓW	42
4.1.1	INFORMACJE OGÓLNE.....	42
4.1.2	ELEMENTY OPISU FORMATU	43
4.1.3	PLIKI POLIMORFICZNE.....	46
4.1.4	KOLEJNOŚĆ ANALIZOWANIA ELEMENTÓW FORMATU	46
4.1.5	DEFINIOVANIE FORMATU W PLIKU IMPEXP.INI.....	46
4.1.6	DEFINIOVANIE FORMATU W SZABLONIE	48
4.1.7	KIEDY DEFINIOWAĆ WŁASNE FORMATY DANYCH.....	48
4.2	PREDEFINIOVANE FORMATY	48
4.2.1	UWAGI OGÓLNE	48
4.2.2	FORMAT SDF	49
4.2.3	FORMAT AM	50
4.2.4	FORMAT CDF	50
4.2.5	FORMAT AMS.....	51
4.2.6	FORMAT POLKA.....	54
4.2.7	FORMAT SUBIEKT	55
4.2.8	FORMAT COMMA.....	55
4.2.9	FORMAT SYMFONIA E-FAKTURY Z EDS	56
5	Elementy języka szablonów	56
5.1	UWAGI OGÓLNE	56
5.2	PRZETWARZANIE DANYCH PRZY POMOCY SZABLONU	56
5.2.1	WCZYTANIE PORCJI DANYCH WEJŚCIOWYCH	56
5.2.2	IDENTYFIKACJA PROCEDURY PRZETWARZANIA	56
5.2.3	TWORZENIE SEKCJI WYJŚCIOWEJ	57
5.2.4	PRZETWORZENIE DANYCH Z SEKCJI WEJŚCIOWEJ.....	57
5.2.5	PODSEKCJE W DANYCH WEJŚCIOWYCH.....	58
5.2.6	PRZETWARZANIE WIELU SEKCJI O TEJ SAMEJ NAZWIE	59
5.3	KOMENTARZE	59
5.4	DEFINICJE	59
5.5	POLECENIA GLOBALNE	59
5.6	POLECENIA SEKCJI.....	59
5.7	SEKCJE SZABLONU.....	60
5.8	POLA WEJŚCIOWE	60
5.9	ZMIENNE	60
5.10	POLA WYJŚCIOWE	60
5.11	ARGUMENTY BEZPOŚREDNIE	60
5.12	OPIS SZABLONÓW XSL.....	60
6	Spis definicji	61

7	Spis operatorów	62
7.1	PODSTAWIENIE	62
7.2	NEGACJA.....	63
8	Spis poleceń	63
8.1	POLECENIA GLOBALNE	63
8.2	POLECENIA SEKCJI.....	64
9	Wzorcowe szablony importu specjalnego.....	69
9.1	SZABLONY AKTUALNIE WYKORZYSTYWANE	69
9.1.1	FK2005.AMS.....	69
9.1.2	SRTRW.AMS	69
9.1.3	PLACE.AMS.....	70
9.1.4	HMFK.AMS	70
9.1.5	HMFFK.AMS	80
9.1.6	EFAKTURY.AMS	80
9.1.7	KIEDY DEFINIOWAĆ WŁASNE SZABLONY IMPORTU?	80
9.1.8	KIEDY DEFINIOWAĆ WŁASNE SZABLONY XSLT?.....	80
9.2	SZABLONY HISTORYCZNE	80
9.2.1	SYM200.AMS.....	80
9.2.2	POLKA.AMS	81
9.2.3	CHART.AMS	81
9.2.4	SUBIEKT.AMS	81
9.2.5	CDN_FPP.AMS.....	81
9.2.6	CDN_FPPK.AMS	82
10	Specyfikacja sekcji wyjściowych dla programu Finanse i Księgowość.....	82
10.1	SPECYFIKACJA PÓL DLA SEKCJI „DOKUMENT”	82
10.2	SPECYFIKACJA PÓL DLA SEKCJI „ZAPIS”	83
10.3	SPECYFIKACJA PÓL DLA SEKCJI „REJESTR”	85
10.4	SPECYFIKACJA PÓL DLA SEKCJI „TRANSAKCJA”	86
10.5	SPECYFIKACJA PÓL DLA SEKCJI „ROZLICZENIE”	87
10.6	SPECYFIKACJA PÓL DLA SEKCJI „KONTRAHENT”.....	87
10.7	SPECYFIKACJA PÓL DLA SEKCJI „INCYDENTALNY”	88
10.8	SPECYFIKACJA PÓL DLA SEKCJI „PRACOWNIK”.....	88
10.9	SPECYFIKACJA PÓL DLA SEKCJI „URZAD”	89
10.10	SPECYFIKACJA PÓL DLA SEKCJI „RACHUNEK”	90
11	Zasady wypełniania walut dla importu specjalnego.	90
11.1	DOKUMENTY BEZWALUTOWE	90
11.2	DOKUMENTY JEDNOWALUTOWE	90
11.3	DOKUMENTY WIELOWALUTOWE.....	91
12	Przykłady szablonu i danych.....	92

12.1	ILUSTRACJA PROCESU PRZETWARZANIA DANYCH	92
12.2	PROCES PRZETWARZANIA EDS Z WYKORZYSTANIEM XSLT	93
12.3	PRZYKŁAD OBSŁUGI DOKUMENTÓW WALUTOWYCH	94
13	Kody błędów importu specjalnego	96
13.1	KODY BŁĘDÓW ZWIĄZANE Z SZABLONEM.....	96
13.1.1	(1) ZBYT DŁUGI SYMBOL:	96
13.1.2	(2) ZBYT DŁUGIE WYRAŻENIE STRINGOWE.....	96
13.1.3	(3) BRAK ZAMYKAJĄCEGO CUDZYSŁOWU.	96
13.1.4	(4) PUSTA LUB ZBYT DŁUGA NAZWA POLA.....	96
13.1.5	(5) NIEZNANE POLECENIE:	96
13.1.6	(6) OCZEKIWANY ZNAK POCZĄTKU BLOKU.	96
13.1.7	(7) OCZEKIWANY PRZECINEK LUB NAWIAS.....	96
13.1.8	(8) OCZEKIWANY NAWIAS ZAMYKAJĄCY.	96
13.1.9	(9) ZIGNOROWANO WYRAŻENIE POZA SEKCJĄ.	96
13.1.10	(10) BŁĄD OTWARCIA PLIKU SZABLONU:	97
13.1.11	(11) NIEOCZEKIWANY KONIEC PLIKU.....	97
13.1.12	(12) BRAK ZNAKU RÓWNOŚCI.....	97
13.1.13	(13) NIEROZPOZNANY SYMBOL:	97
13.1.14	(14) OCZEKIWANE POLECENIE PROSTE.....	97
13.1.15	(15) NIEZNANY TYP POLECENIA:	97
13.1.16	(16) OCZEKIWANY NAWIAS OTWIERAJĄCY.	97
13.1.17	(17) OCZEKIWANY PRZECINEK.	97
13.1.18	(19) POLE '...' NIE ZOSTAŁO ZNALEZIONE.....	97
13.1.19	(20) ARGUMENT NIE ZAWIERA LICZBY.	97
13.1.20	(21) WYNIK JEST POZA DOPUSZCZALNYM ZAKRESEM.....	97
13.1.21	(22) PUSTA NAZWA ZMIENNEJ LUB DEFINICJI.	97
13.1.22	(23) DEFINICJA NIE MOŻE BYĆ UMIESZCZONA WEWNĄTRZ SEKCJI LUB POLECENIA.....	97
13.1.23	(24) OCZEKIWANA WARTOŚĆ TEKSTOWA.	98
13.1.24	(25) OCZEKIWANE 'CASE'.	98
13.1.25	(26) BŁĘDNIE UMIESZCZONY ZNAK:	98
13.1.26	(27) ZBYT WIELE OTWARTYCH SZABLONÓW.	98
13.1.27	(27) NIEZNANE POLECENIE GLOBALNE:	98
13.1.28	(28) SEKCJA WYJŚCIOWA NIE ZOSTAŁA UTWORZONA.	98
13.1.29	(29) POMINIĘTO SEKCJĘ:	98
13.1.30	(30) BRAK OTWARTEJ SEKCJI WEJŚCIOWEJ.....	98
13.1.31	(31) NIEPOPPRAWNA LUB NIEZGODNA SPECYFIKACJA FORMATU DATY.	98
13.1.32	(32) DZIELENIE PRZEZ ZERO.	98
13.1.33	(33) PO ZNAKU + LUB - OCZEKIWANA JEST LICZBA.	98
13.1.34	(34) POLECENIE #BASIC MOŻE WYSTĄPIĆ TYLKO RAZ.....	98

13.1.35	(200) BŁĄD WEWNĘTRZNY, ZŁY NUMER POLECENIA: ...	98
13.1.36	(201) ZŁY TYP POLECENIA.	98
13.1.37	(202) BŁĄD WEWNĘTRZNY, ZŁY TYP ARGUMENTU: ...	98
13.2	KODY BŁĘDÓW ZWIĄZANE Z PLIKIEM WEJŚCIOWYM	99
13.2.1	(300) NIEROZPOZNANA SEKCJA WEJŚCIOWA:	99
13.2.2	(400) W PLIKU DANYCH NIE ODNALEZIONO KOŃCA KOMENTARZA BLOKOWEGO.	99
13.2.3	(401) W PLIKU DANYCH ZNALEZIONO POLE Z PUSTĄ WARTOŚCIĄ.	99
13.2.4	(402) PLIK Z DANYMI NIE ZOSTAŁ JESZCZE OTWARTY.	99
13.2.5	(403) W PLIKU DANYCH NIE MA ŻADNEGO WPISU.	99
13.2.6	(404) PLIK Z DANYMI NIEDOSTĘPNY. NIE ISTNIEJE LUB JEST UŻYWANY PRZEZ KOGOŚ INNEGO. .	99
13.2.7	(405) PLIK Z DANYMI JEST JUŻ OTWARTY.	99
13.2.8	(406) NIE PODANO FORMATÓW SEKCJI I PLIKU.....	99
13.2.9	(407) NIEROZPOZNANY IDENTYFIKATOR TYPU PLIKU W SZABLONIE.	99
13.2.10	(408) NIEPOPRAWNY ELEMENT FORMATU SEKCJI LUB PLIKU W SZABLONIE.	99
13.2.11	(409) ZNALEZIONO W PLIKU ELEMENT NIE ODPOWIADAJĄCY ŻADNEMU WZORCOWI.....	99
13.2.12	(410) ZBYT DUŻO ZAGNIEŹDŻONYCH PODSEKCJI.	99
13.2.13	(411) NIE PODANO LISTY SZEROKOŚCI PÓL FORMATU CDF.	99
13.2.14	(412) BŁĄD W WYRAŻENIU REGULARNYM.	100
13.2.15	(413) NIEPOPRAWNE FORMATY SEKCJI W SZABLONIE.	100
13.3	KODY BŁĘDÓW ZWIĄZANE Z KONTROLĄ POPRAWNOŚCI	100
13.3.1	(600) NIEPRAWIDŁOWA IDENTYFIKACJA ZAPISÓW NA KONTO KASY.....	100
13.3.2	(601) W DEFINICJI UŻYTEGO REJESTRU NIE MA WYBRANEJ STAWKI	100
13.3.3	(602) ZAPIS NR ... NIE ODNALEZIONO TRANSAKCJI DO ROZLICZENIA. ROZLICZENIE ZASTĄPIŁO NOWĄ TRANSAKCJĄ.	100
13.3.4	(603) NIEZGODNOŚĆ RODZAJU ZAKUPÓW (ABC) POMIĘDZY DANYMI REJESTRU I JEGO DEFINICJĄ.....	100
13.3.5	(604) BRAK ZAPISÓW W DOKUMENCIE.	100
13.3.6	(605) NIE ODNALEZIONO TRANSAKCJI DO ROZLICZENIA. ROZLICZENIE ZASTĄPIŁO NOWĄ TRANSAKCJĄ.....	100
13.3.7	(607) DOKUMENT: ... WYSTĄPIŁ BŁĄD PODCZAS ZAPISYWANIA.	101
13.3.8	(608) BRAK DEFINICJI REJESTRU UŻYTEGO W UZGODNIENIU.....	101
13.3.9	(609) BŁĄD WYKONANIA PROCEDURY AMBASIC.	101
13.3.10	(610) BRAK POLA LUB PUSTA WARTOŚĆ POLA 'KLUCZ' W UZGADNIANEJ SEKCJI.....	101
13.3.11	(611) NIE ZNALEZIONO ZAPISU ODPOWIADAJĄCEGO SEKCJI ROZRACHUNKÓW.	101
13.3.12	(612) NIEISTNIEJĄCE LUB NIEPRAWIDŁOWE KONTO DLA ROZRACHUNKÓW. ROZRACHUNKI ZOSTAŁY ZIGNOROWANE.....	101
13.3.13	(613) ZAPIS NR ... NIEISTNIEJĄCE LUB NIEPRAWIDŁOWE KONTO DLA ROZRACHUNKÓW. ROZRACHUNKI ZOSTAŁY ZIGNOROWANE.	101
13.3.14	(614) REJESTR ... MA OKRES PRZYSZŁOROCZNY, JEDNAK W PRZYSZŁYM ROKU NIE ISTNIEJE ZGODNA DEFINICJA REJESTRU.....	101
13.3.15	(615) REJESTR VAT NIE MA OKREŚLONEGO RODZAJU ZAKUPU ABC	101

13.3.16	(616) W REJESTRZE VAT NIE MA PODANEJ STAWKI VAT.	101
13.3.17	(617) NIEKOMPLETNE DANE REJESTRU VAT.	102
13.3.18	(618) POMINIĘTE NIE PEŁNE INFORMACJE O ROZLICZENIU W RÓŻNYCH WALUTACH	102
13.3.19	(619) DO (623)	102
13.3.20	(624) WYMUSZENIE DATY OKRESU.....	102
14	Odpowiedzi na najczęściej zadawane pytania	102

1 Wstęp

1.1 Informacje ogólne

Import specjalny jest uniwersalnym mechanizmem umożliwiającym wczytywanie do programu Finanse i Księgowość danych z innych programów. Za pomocą importu specjalnego mogą być wczytywane dane do wszystkich kartotek za wyjątkiem słowników, oraz wszystkie typy dokumentów księgowych, wraz z dekretacjami, danymi do rejestrów VAT oraz rozrachunkami. Importowane dokumenty są umieszczane w buforze programu. Dzięki zastosowaniu mechanizmu konfigurowania, a raczej opisywania danych wejściowych można wczytywać dane w różnych formatach oraz o różnej konstrukcji logicznej. Dzięki swej elastyczności import specjalny zapewnia wymianę danych pomiędzy modułami Symfonii, Systemu zarządzania ERP, a także współpracę z programami innych producentów. W ofercie usług Sage jest również pomoc i tworzenie importów na zamówienie. Każda seryjna wersja programu Symfonia Finanse i Księgowość posiada kompletny zestaw narzędzi i mechanizmów potrzebnych do stworzenia dodatkowego własnego formatu wymiany danych.

Mechanizmy zawarte w imporcie specjalnym umożliwiają dowolną obróbkę danych wejściowych i przetworzenie ich do postaci wymaganej przez program. Możliwe jest m.in. wstępne przetwarzanie całego pliku danych (np. unzip), zmiana kolejności pól, zmiana formatu daty lub kwoty, łączenie lub dzielenie zawartości pól, wyliczanie, operacje warunkowe oraz wiele innych. Od wersji 2012 import specjalny możliwy jest z plików XML. Wstępne przetwarzanie XML do formatu tekstowego następuje za pomocą szablonu XSLT.

Niniejszy plik zawiera informacje zgodne z wersją 2018.2.

1.2 Formaty danych

1.2.1 Struktura danych

Import specjalny umożliwia wczytywanie wyłącznie danych w postaci tekstowej. Dane wejściowe mogą być w zapisane w jednym z poniższych formatów:

AMS

Format wymiany danych pomiędzy modułami systemu Symfonia, stosowany od wersji 3.0 programu Finanse i Księgowość. Format ten został wprowadzony wraz z ówczesną rozbudową możliwości importu specjalnego. Jest to format danych z nazwami pól, grupujący dane w nazwane sekcje. Jest to obecnie preferowany format danych wejściowych. W tym formacie zarówno ogólna ilość jak i kolejność występowania poszczególnych danych nie ma znaczenia. Dane są rozpoznawane poprzez ich nazwę, a nie położenie w pliku danych.

SDF

Pola o zmiennej długości, oddzielone od siebie średnikiem jako separatorem. Jeden wiersz danych zakończonych znakiem końca linii jest traktowany jako jedna porcja danych, przetwarzanych jako całość. W tym formacie istotna jest kolejność danych w wierszu.

UWAGA: Import specjalny akceptuje dane, w których w jednym pliku wiersze mają różną ilość pól danych.

CDF

Pola o stałej długości. Jeśli pole zawiera mniejszą liczbę znaków, wówczas jest dopełniane do określonego rozmiaru spacjami. Jeden wiersz danych zakończonych znakiem końca linii jest traktowany jako jedna porcja danych, przetwarzanych jako całość. W tym formacie istotna jest kolejność danych w wierszu.

COMMA

Pola o zmiennej długości, oddzielone przecinakami jako separatorami. Jeden wiersz danych zakończonych znakiem końca linii jest traktowany jako jedna porcja danych, przetwarzanych jako całość. W tym formacie istotna jest kolejność danych w wierszu.

UWAGA: Import specjalny akceptuje dane, w których w jednym pliku wiersze mają różną ilość pól danych.

AM

Format importu z innych modułów systemu, stosowany w wersji 2.0 programu Finanse i Księgowość. Jest pewną modyfikacją formatu SDF. W tym formacie istotna jest zarówno kolejność pól w wierszu jak też logiczne połączenie i kolejność poszczególnych wierszy danych, tworzących jeden dokument

POLKA

Format danych stosowany podczas importu danych z programu POLKA. Jest to format z nazwami pól, dodatkowo grupujący dane w postaci nazwanych sekcji.

SUBIEKT

Format wymiany danych używany przez program Subiekt firmy Insert

Import specjalny posiada mechanizmy umożliwiające zdefiniowanie innego, własnego formatu danych różnego od podanych powyżej predefiniowanych formatów. Dzięki tym zaawansowanym właściwościom możliwe jest przystosowywanie programu do wczytywania danych z innych programów bez konieczności dokonywania zmian w samym programie.

XML

Uniwersalny format wymiany danych w formacie hierarchicznym. Format został zastosowany do importu e-Faktury.

Import specjalny wykorzystuje szablon XSLT do zamiany XML na format AMS, a następnie szablon AMS do wczytania danych.

1.2.2 Kodowanie znaków

Import specjalny zapewnia obsługę dowolnego sposobu kodowania polskich znaków. Wbudowane formaty kodowania są następujące:

LATIN2

MAZOWIA

W imporcie specjalnym można ponadto zdefiniować dowolny dodatkowy, własny sposób kodowania znaków. Polega to na zdefiniowaniu własnej tablicy przekodowania znaków. W ten sposób można również realizować także proste mechanizmy ukrywania lub zniekształcania wartości danych w celu uniemożliwienia ich odczytania przez osoby niepowołane (np. zamiana cyfr w kwotach, liter w opisach itd.).

Import danych w formacie XML pozwala wczytać dane w dowolnym formacie. Dane wyjściowe w formacie AMS zapisywane są w standardowym CP-1250.

1.2.3 Pola daty

W danych wejściowych może być stosowany dowolny sposób zapisywania daty. Dotyczy to zarówno długości poszczególnych elementów daty (rok, miesiąc, dzień) jak też ich kolejności oraz separatora oddzielającego od siebie kolejne elementy. Możliwy jest zarówno pełny jak i skrócony zapis roku (4 lub 2 cyfry)

1.2.4 Pola kwoty

Kwoty mogą być zapisane z wykorzystaniem dowolnego znaku jako separatora dziesiętnego oraz tysięcznego. Dla potrzeb importu specjalnego należy określić tylko znak odpowiadający separatorowi dziesiętnemu (oddzielającemu część całkowitą od ułamkowej). Wszystkie znaki różne od znaku liczby (+lub -), cyfr oraz separatora dziesiętnego są ignorowane (pomijane).

1.2.5 Pola złożone

Import specjalny umożliwia obsługę pól, które w pliku danych stanowią jedną wartość, a zawierają w sobie informacje różnego rodzaju. Przykładem może być podanie w jednym polu danych nazwy ulicy razem z numerem domu. W programie Finanse i Księgowość informacje te powinny być rozdzielone na dwie niezależne informacje: nazwę ulicy i numer domu. Import specjalny posiada mechanizmy umożliwiające rozdzielanie takich informacji, poprzez stosowanie tzw. Wyrażeń regularnych, opisujących konstrukcję logiczną takich połączonych informacji.

1.2.6 Pola składane

Import specjalny umożliwia dowolne łączenie informacji z wielu pól danych w jedną całość, przykładem może być tworzenie numeru konta przez połączenie określonego numeru syntetyki z numerem kontrahenta jako pierwszym poziomem analityki.

1.3 Pojęcia podstawowe

Poniżej przedstawione są pojęcia związane z importem specjalnym, danymi wejściowymi oraz mechanizmami przekształcania danych.

Pole (danych)

Jest to elementarna, informacja w pliku danych, opisująca jedną wartość (np. kwota, konto, data, numer, opis itp.). Poniżej pokazano przykładowe trzy pola danych oddzielone od siebie separatorem (średnikiem):

```
2457;Abacki;Janusz
```

Nazwa (pola)

Opcjonalna jawnie podana nazwa pola przechowującego określoną wartość. W zależności od formatu w pliku danych mogą być umieszczone tylko pola danych (bez nazwy), lub też każde pole danych może mieć dodatkowo swoją nazwę.

Poniżej pokazano przykładowe dane z jawnie podaną nazwą pola oraz znakiem równości oddzielającym nazwę i wartość:

```
numer=2457  
nazwisko=Abacki  
imię=Janusz
```

Sekcja wejściowa

Jest to elementarna porcja danych (grupa), zawierająca komplet informacji (pól) niezbędnych do jej przetworzenia i wczytania do programu importującego dane. Składa się z pól oraz z podsekcji. Przykładem sekcji jest informacja o całym jednym dokumencie lub kontrahencie. Dla formatów SDF, CDF, COMMA jest to jeden wiersz z pliku wejściowego, dla pozostałych formatów jest to grupa kolejnych linii tworzących komplet danych. Poniżej podano przykładowe trzy sekcje (w tym przypadku jedna sekcja to jedna linia) z pliku w formacie SDF, zawierające dane pracowników:

```
2457;Abacki;Janusz  
2458;Kowalski;Maciej  
2459;Zawadzki;Zenon
```

Podsekcja

Jest to logicznie wydzielona grupa informacji w ramach sekcji. Składa się z pól oraz ew. z dalszych podsekcji. Przykładem podsekcji może być informacja o kwotach w rejestrze VAT lub informacje o jednym zapisie na koncie. Podział sekcji danych na podsekcje jest często umowny. Poniżej pokazano podsekcję zapisu w sekcji dokumentu, w formacie AMS

```
Dokument{  
  data=12-02-2000  
  Zapis{  
    kwota=234,56  
    konto=201-1  
  }  
}
```

Plik wejściowy

Jest to plik na dysku, zawierający dane importowane. każdy plik danych zawiera szereg powtarzających się informacji (sekcji), opisujących np. kolejne importowane dokumenty. Każdy plik danych jest czytany sekwencyjnie, sekcja po sekcji od początku do końca. W pewnych przypadkach wszystkie informacje wczytywane w jednej operacji importu mogą być zawarte w kilku plikach danych.

Szablon

Jest to program napisany w specjalnym języku programowania, dzięki któremu dokonywana jest interpretacja danych wejściowych i zamiana do postaci wymaganej przez program odbierający dane. Dzięki szablónowi dane wejściowe które mogą być zapisane na wiele różnych sposobów, zamieniane są do postaci akceptowalnej przez program odbierający dane. W najprostszym wariancie szablon zawiera przypisania (przyporządkowania) kolejnych pól wejściowych do odpowiednich pól w programie. W bardziej złożonych przypadkach w szablonie znajdują się dodatkowo wyrażenia logiczne, arytmetyczne lub inne złożone operacje na danych.

Szablon XSLT

Jest to program napisany w języku XSLT, dzięki któremu dokonywana jest transformacja danych wejściowych w formacie XML i zamiana danych do postaci AMS, które następnie przy pomocy Szablónu są odbierane przez program. Dzięki szablónowi XSLT dane wejściowe mogą być zapisane w dowolnym formacie XML. W szablonie XSLT używany jest ustandaryzowany język XPath do dokonywania prostych transformacji i mapowań.

Sekcja wyjściowa

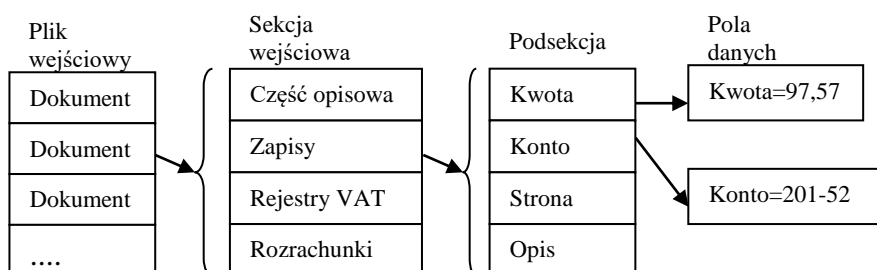
Zakresem danych i konstrukcją logiczną odpowiada informacjom w bazie danych programu Finanse i Księgowość. Jest to elementarna porcja danych, utworzona przez wczytanie i przetworzenie jednej sekcji wejściowej danych. Postać sekcji wyjściowej jest ściśle ustalona, zgodna z wbudowanymi procedurami kontroli poprawności oraz zapisywania danych do baz danych programu. Sekcje wyjściowe tworzy się jawnie poprzez wykonanie odpowiednich poleceń w szablonie. Pola danych w sekcji wyjściowej tworzy się przez wykonanie odpowiednich poleceń podstawienia danych.

Postać sekcji wyjściowej jest zależna od rodzaju wczytywanych danych. Inna jest dla kartotek, inna dla poszczególnych typów dokumentów.

UWAGA: Sekcje wyjściowe są tworzone przez szablon. Można w ten sposób utworzyć absolutnie dowolną strukturę sekcji i podsekcji wyjściowych, oraz ich pól danych. Aby jednak dane wyjściowe nie zostały przez program odrzucone na etapie kontroli poprawności, szablon musi być tak zaprojektowany, aby stworzyć właściwą strukturę tych danych, zgodną ze specyfikacją programu Finanse i Księgowość. W przyszłości mechanizm importu specjalnego może być zastosowany w innych programach Systemu Symfonia, wówczas wszystkie opisane tu elementy i mechanizmy pozostaną niezmiennione, inna będzie tylko postać danych wyjściowych, wymaganych i akceptowanych przez program odbierający dane.

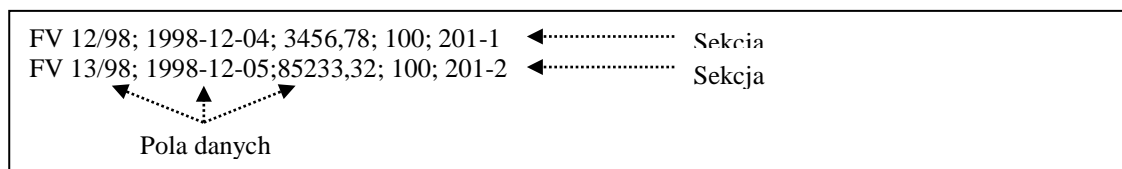
Uzgodnienia

Proces łączenia informacji z pliku danych z informacjami zawartymi w programie Finanse i księgowość. Przykładowo każdy oddzielny typ dokumentu importowanego należy połączyć z właściwym, odpowiadającym mu typem dokumentu w programie. Każde raz dokonane uzgodnienie jest pamiętane przez program i wykorzystywane podczas importowania kolejnych danych.



Rysunek 1 Elementy danych importowanych

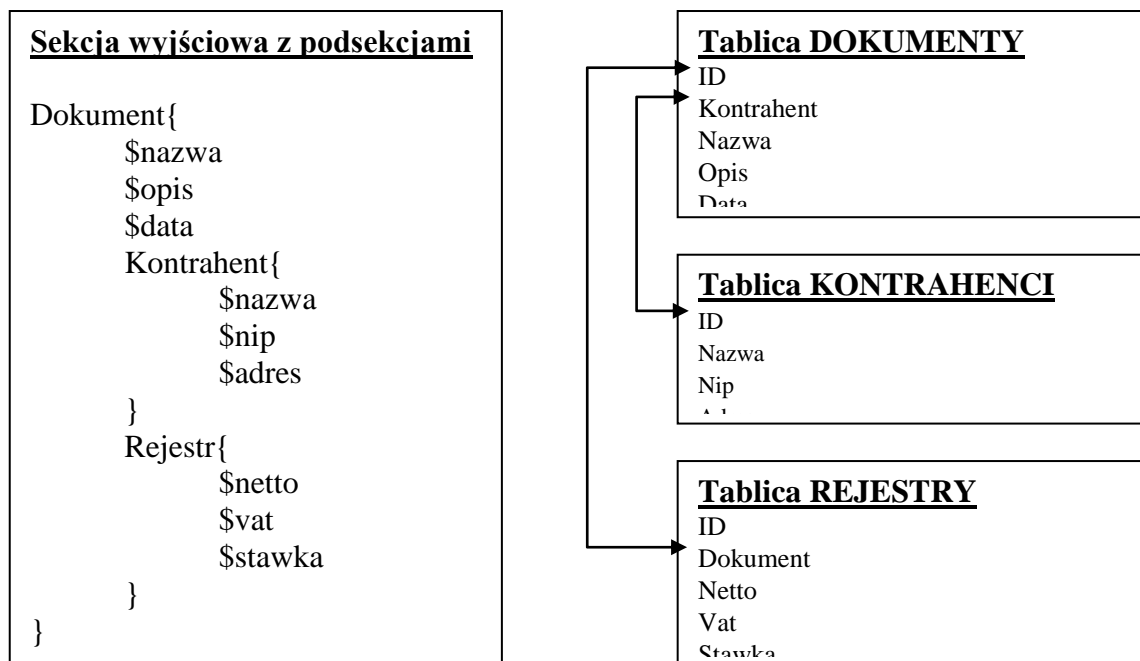
Plik wejściowy SDF z zaznaczeniem podstawowych elementów danych



1.4 Sekcje wyjściowe

1.4.1 Informacje ogólne

Sekcje wyjściowe są odpowiednikiem baz danych programu. Są miejscem, gdzie wpisywane są przetworzone dane wejściowe. Zawartość sekcji wyjściowej jest poddawana procesowi uzgadniania danych, kontroli poprawności i uzupełnienia danych. Po prawidłowym zakończeniu tych procesów dane z sekcji wyjściowej są zapisywane do baz danych programu. Struktura logiczna i zawartość informacyjna sekcji wyjściowych odpowiada bezpośrednio strukturalnym bazom danych programu Finanse i Księgowość. Dane wejściowe zawierają informacje trafiające do różnych tablic w bazie danych programu (np. dokumenty, zapisy, rejestry VAT, kontrahenci itp.), ale powiązane ze sobą ścisłymi powiązaniem (relacjami). Przykładowo zapisy i rejestry są związane z konkretnym dokumentem, faktura VAT dotyczy konkretnego kontrahenta itp. W pliku wyjściowym powiązania te są zapewnione przez umieszczenie wszystkich danych w jednej sekcji (w formacie SDF, CDF, COMMA) lub w postaci podsekcji umieszczonych w jednej sekcji (formaty AM, AMS, POLKA itp.). Powiązania (relacje) pomiędzy poszczególnymi danymi przechowywanymi docelowo w różnych tablicach baz danych programu muszą być zachowane również w sekcji wyjściowej. Z tego względu wygodniej jest myśleć o sekcji wyjściowej nie jako o zbiorze oddzielnych tablic, ale jak o strukturze analogicznej jak sekcje wejściowe, czyli o sekcjach i zawartych w nich podsekcjach. Dzięki zastosowaniu „zagnieżdżania” podsekcji wyjściowych można zapomnieć o wzajemnych relacjach pomiędzy tablicami bazy danych, relacje są tworzone automatycznie, na podstawie wzajemnego umiejscowienia podsekcji w sekcji wyjściowej.



Rysunek 2 porównanie organizacji sekcji wyjściowej i relacji pomiędzy tablicami w bazie danych

Istotne dla prawidłowego przebiegu importu danych jest utworzenie właściwej hierarchii podsekcji w sekcji wyjściowej. Istnieje wiele dopuszczalnych kombinacji wzajemnego umieszczania podsekcji. Od tego umiejscowienia zależy ostateczna postać wczytanych danych. Najprostszym przypadkiem jest pojedyncza sekcja, nie posiadająca podsekcji. Taka sytuacja może mieć miejsce podczas importu samej kartoteki kontrahentów. W takim przypadku każdy dodawany rekord umieszczany jest tylko w kartotece kontrahentów, bez żadnych dodatkowych powiązań. W przypadku dokumentów pojawi się w sekcji wyjściowej co najmniej jedna sekcja rejestru albo dwie lub więcej sekcji zapisów. Dodatkowo mogą pojawić się sekcje transakcji, rozliczeń oraz kartotek.

1.4.2 Pola danych

Każda sekcja wyjściowa składa się z pól danych oraz ewentualnie może zawierać podsekcje. O ile sekcja odpowiada jednej tabeli w bazie danych programu, o tyle pola sekcji odpowiadają kolejnym kolumnom (polom) w danej tabeli bazy danych.

Pola sekcji składają się z nazwy rozpoczynającej się znakiem \$, oraz dowolnej wartości.

Tworząc szablon importu specjalnego można utworzyć dowolną ilość sekcji oraz pól wyjściowych o dowolnych nazwach i wartościach. Wśród tych wszystkich sekcji i pól wyjściowych program będzie analizował tylko te, których nazwy są mu znane. Pozostałe pola i sekcje zostaną zignorowane bez ostrzeżenia. Lista sekcji i pól które będą zrozumiałe dla programu została zamieszczona w specyfikacji sekcji i pól wyjściowych.

Takie rozwiązanie ma swoje wady i zalety. Oczywiście wadą jest fakt, że program nie „wyłapie” przypadkowych błędów w nazwach pól lub sekcji (o ile nie są to pola krytyczne dla przebiegu importu danych). Zaletą jest to, że import specjalny nie jest wrażliwy na zmiany formatu i struktur danych w programie Finanse i Księgowość.

Część pól danych w każdej sekcji jest obowiązkowa, to znaczy pola te muszą zostać wypełnione, aby import danych mógł przebiegać prawidłowo. Część pól może być wypełniona lub pozostać pusta. Pola te zwykle mogą być automatycznie uzupełnione przez program na podstawie wartości innych pól, lub też ich obecność nie jest obowiązkowa (np. pole uwag dla kontrahenta, treść dokumentu itp.). Istnieją takie pola, które nie mogą być wypełnione podczas importu specjalnego. Wypełnienie ich niedozwolonymi wartościami może spowodować błędy w importowanych danych. Wszystkie pola wraz z ich charakterystyką zostały opisane w specyfikacji pól wyjściowych.

1.4.3 Pole \$klucz

Każda sekcja związana z kartoteką programu musi zawierać pole **\$klucz**, które jest używane do identyfikacji i połączenia danych od konkretnego nadawcy, z danymi programu FK. Wartość tego pola jednoznacznie identyfikuje informację od nadawcy.

Pole \$klucz nie może być puste, bo spowoduje to wyświetlenie komunikatu i natychmiastowe przerwanie importu danych. , jeśli dla kartoteki kontrahentów pole \$klucz będzie zawierać wartość „0”, to sekcja kontrahenta zostanie zignorowana (pominięta) bez żadnego komunikatu.

Pole \$klucz dla kartotek (kontrahenci, pracownicy, urzędy itp.) musi identyfikować jednoznacznie pozycję kartoteki u nadawcy. Może to być np. numer w kartotece nadawcy lub unikalny skrót. Dla pracownika może to być np. numer PESEL

UWAGA: Wartość pola \$klucz musi być unikalna dla każdej pozycji kartoteki. Przykładowo numer NIP może być niewystarczającą informacją, gdyż kilka oddziałów jednej firmy (a więc kilka pozycji w kartotece kontrahentów) może mieć ten sam numer NIP-u. Najlepszą informacją dla pola \$klucz jest zawsze numer pozycji w kartotece nadawcy

Pole \$klucz dla dokumentów określa typ dokumentu wzorcowego w FK, jaki ma zostać użyty podczas zapisywania dokumentu do bufora. Zasada jest następująca: Pojawienie się każdej nowej wartości pola \$klucz powoduje otwarcie okna uzgodnień dokumentów i użytkownik musi mu przypisać jakiś typ dokumentu zdefiniowanego w programie FK. Każdy następny importowany dokument, posiadający takie same pole \$klucz zostanie przypisane do tego wybranego typu dokumentu. Przykładowo jeśli dokument będzie miał wartość pola \$klucz równą np. „F/R00” i użytkownik połączy ten dokument z wzorcowym dokumentem FVS, wówczas ten dokument oraz wszystkie następne dokumenty posiadające taką samą wartość klucza zostaną zapisane do bufora jako dokumenty typu FVS.

UWAGA: Różne wartości pola \$klucz mogą zostać przypisane do tego samego dokumentu wzorcowego w programie FK. Wszystko zależy od decyzji księgowego, dokonującego uzgodnień typów dokumentów podczas importu specjalnego.

Dla raz opracowanego szablonu współpracującego z ustaloną postacią pliku danych wejściowych, pole \$klucz nie powinno już podlegać modyfikacji. Jego zmiana spowodowałaby konieczność uzgadniania na nowo uzgodnień pomiędzy nadawcą i odbiorcą, w skrajnych przypadkach mogłaby również doprowadzić do błędnej interpretacji powiązań i w rezultacie skojarzyć dane w niewłaściwy sposób.

Klucz może być dowolnym ciągiem znaków o długości do 100 znaków. Pole \$klucz jest generowane przez szablon na podstawie danych wejściowych i nie musi w sposób jawny występować w pliku danych.

Sekcje które nie wymagają uzgodnień nie wymagają wypełniania pola \$klucz. Są to następujące sekcje:

ZAPIS

REJESTR

TRANSAKCJA

ROZLICZENIE

Sekcje te nie mogą występować samodzielnie (jako sekcje główne), lecz mogą być użyte wyłącznie jako podsekcje w sekcji DOKUMENT

1.4.4 Dozwolone kombinacje sekcji i podsekcji

Poniżej przedstawiono dozwolone kombinacje wzajemnych powiązań sekcji i podsekcji w programie Finanse i Księgowość.

Pozycja w kartotece kontrahentów stałych, bez powiązań z innymi danymi w programie.

```
Kontrahent {  
    $klucz =  
    //reszta pól kartoteki kontrahentów  
}
```

Pojawienie się takiej sekcji powoduje utworzenie nowej pozycji w kartotece kontrahentów (o ile nie istnieje w programie Finanse i Księgowość), lub powiązanie kontrahenta z pliku danych z istniejącym kontrahentem w kartotece. Zarówno wprowadzenie nowej pozycji, jak też uzgodnienie z istniejącym w kartotece może wymagać dodatkowej operacji wykonanej przez użytkownika. Sposób rozstrzygania o utworzeniu nowej pozycji i kryteria przyjmowane do uznania zgodności z istniejącymi pozycjami kartotek, zmieniły się istotnie od wersji 2009 i są opisane w osobnym dokumencie. Ponadto skrótowy opis jest przedstawiony w podręczniku i pomocy do programu Finanse i Księgowość – dotyczy to także kartoteki pracowników.

Pozycja w kartotece kontrahentów incydentalnych, bez powiązań z innymi danymi w programie

```
Incydentalny {  
    $klucz =  
    //reszta pól kartoteki kontrahentów incydentalnych  
}
```

Pozycja w kartotece pracowników, bez powiązań z innymi danymi w programie

```
Pracownik {  
    $klucz =  
    //reszta pól kartoteki pracowników  
}
```

Pozycja w kartotece urzędów skarbowych, bez powiązań z innymi danymi w programie

```
Urzad {  
    $klucz =  
    //reszta pól kartoteki urzędów  
}
```

Pozycja w kartotece rachunków bankowych, bez powiązań z innymi danymi w programie

```
Rachunek {  
    $klucz =  
    //reszta pól kartoteki rachunków bankowych  
}
```

Dokumenty w buforze.

```

Dokument {
    $klucz =
    //reszta pól części opisowej dokumentu
    Kontrahent { //o ile występuje na dokumencie
//oprócz sekcji kontrahent mogą również
//wystąpić pozostałe kartoteki programu
        $klucz=
        //reszta pól kontrahenta
    }
    Rejestr { //o ile występuje w dokumencie
        //może być powtórzony wiele razy
        $klucz=
        //dane rejestru
    }
    Zapis { //powtórzony tyle razy, ile jest dekrétów
        //nie ma pola klucz
        //pola zapisu
        Transakcja { //o ile jest transakcja dla tego zapisu
            //specyfikacja nowej transakcji dla danego zapisu
            //nie ma pola klucz
            //dane transakcji
        }
        Rozliczenie { //o ile jest rozliczenia dla tego zapisu
            //specyfikacja rozliczenia transakcji dokonanego
            //danym zapisem
            //nie ma pola klucz
            //dane rozliczenia
        }
    }
}
Transakcja {
    //nie ma pola klucz
    //dane transakcji
}
Rozliczenie {
    //nie ma pola klucz
    //dane rozliczenia
}
}

```

Sekcja wyjściowa *Dokument* jest najbardziej rozbudowana (posiada największą liczbę możliwych podsekcji). Jest też najistotniejsza z punktu widzenia programu *Finanse i Księgowość*. Podany powyżej przykład obejmuje najszerszy zakres sekcji wyjściowej dokumentu. Dla różnych dokumentów sekcja wyjściowa może mieć inny zestaw podsekcji oraz inny zakres danych (pól wyjściowych w każdej sekcji).

Podstawowe różnice w ilości różnych podsekcji dokumentu wynikają z następujących czynników:

- Występowania kontrahenta w części opisowej dokumentu
- Występowania pozycji kartotek jako numerów kont w zapisach
- Obecności rejestrów VAT w dokumencie
- Występowania rozrachunków związanych z poszczególnymi zapisami
- Występowania rozrachunków związanych z całym dokumentem

1.5 Przebieg procesu przetwarzania danych

Zasadniczym zadaniem importu specjalnego jest zmiana formatu danych wejściowych tak, aby odpowiadały wymogom programu importującego dane. Podczas importu mogą ulec zmianie wszystkie lub tylko część atrybutów danych, m.in.:

- Kolejność danych
- Wartość poszczególnych danych
- Sposób kodowania polskich znaków
- Relacje pomiędzy danymi

Ponadto:

Część danych (nadmiarowych) może zostać odrzucona

Brakujące dane mogą zostać odtworzone z innych danych lub uzupełnione wartościami domyślnymi

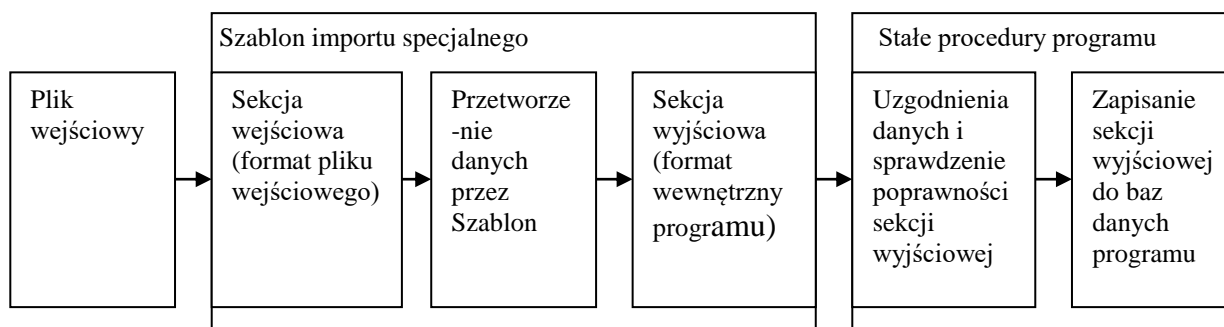
Wszystkie te operacje są zdefiniowane w szablonie importu specjalnego lub w procedurach wbudowanych kontroli poprawności.

Dane wejściowe są przetwarzane cyklicznie w następujących etapach

- czytana jest jedna sekcja z danych wejściowych
- wczytana sekcja (sekcja wejściowa) jest przetwarzana przez szablon do postaci wymaganej przez program i zapisywana do sekcji wyjściowej
- program importujący dokonuje wszystkich potrzebnych uzgodnień danych i sprawdza poprawność sekcji wyjściowej
- dane są zapisywane do baz danych programu

Powyższy ciąg operacji wykonywany jest dla każdej sekcji w danych wejściowych.

Proces wczytywania danych ilustruje poniższy rysunek:



Rysunek 3 Proces importu danych

1.6 Zasady tworzenia sekcji wyjściowych

Najistotniejszym zadaniem prawidłowo skonstruowanego szablonu importu specjalnego jest utworzenie poprawnych sekcji wyjściowych, zawierających właściwe pola danych. Cały proces i wszystkie mechanizmy importu specjalnego dostępne użytkownikowi służą temu celowi. Sekcje wyjściowe są sprawdzane przez wbudowane procedury programu i na ich podstawie zapisywane są informacje do bazy danych (bufora, kartotek itp.) programu Finanse i księgowość.

1.6.1 Kartoteki

Sekcje wyjściowe dla kartotek są najprostszymi sekcjami w imporcie specjalnym. Dla ich prawidłowego wypełnienia wystarczy wypełnić nazwę sekcji, stosownie do importowanej kartoteki, prawidłowo wypełnić pole \$klucz oraz pola wymagane dla danej kartoteki.

\$Klucz

Pole \$klucz musi zawierać informację identyfikującą jednoznacznie każdą pozycję kartoteki u nadawcy. Jeśli dla kilku różnych pozycji w jednej kartotece wystąpiłaby identyczna wartość pola \$klucz, zostałyby błędnie przypisane te pozycje np. do kont lub dokumentów.

Pole \$klucz nie może być puste, bo spowoduje to wyświetlenie komunikatu i natychmiastowe przerwanie importu danych. , jeśli dla kartoteki kontrahentów pole \$klucz będzie zawierać wartość „0”, to sekcja kontrahenta zostanie zignorowana (pominięta) bez żadnego komunikatu.

Przykład:

Import kartoteki kontrahentów z pliku w formacie SDF (pola oddzielone średnikami)

Dane importowane:

```
1;"ABM";"ABM Sp.Cyw."; "LUBLIN"; "CZERWONYCH SZTANDARÓW 94"; "737-080-99-99"
2;"APEX";"PPHU APEX"; "Warszawa"; "ARABSKA 38"; "107-838-09-33"
3;"BESTER";"BESTER"; "KATOWICE"; "KLONOWA 22"; "737-939-00-23"
4;"MIG";"MIG Sp. z o.o."; "W-wa"; "Okopowa78"; "108-23-65-876"
5;"Oświata";"Spółdzielnia Oświata"; "Katowice"; "Mickiewicza 7"; "233-34-56-345"
6;"MBH";"MBH Sp. z o.o."; "W-wa"; "Gen. Zajaczka 7"; "118-00-05-814"
```

Kolejne pola w pliku danych to:

- 1 – numer w kartotece nadawcy
- 2 – skrót nazwy
- 3 – nazwa pełna
- 4 – miasto
- 5 – ulica i numer domu
- 6 - NIP

Przykładowy szablon importu danych:

```
#format „SDF”

Section(„”, „kontrahent”)
{
    $klucz = [1]
    $skrot = [2]
    $nazwa = [3]
    $miejscowosc=[4]
    $ulica = [5]
    $nip = [6]
}
```

W powyższym przykładowym szablonie nazwa sekcji wejściowej jest pusta (pusty cudzysłów jako pierwszy tekst w nawiasie). Wynika to z faktu, że w formacie SDF sekcje wejściowe nie mają nazwy.

Sekcja wyjściowa utworzona przez ten szablon to „kontrahent”, czyli importowane dane zostaną umieszczone w kartotece kontrahentów.

Jako wartość pola \$klucz przyjęto tutaj pierwsze pole z pliku, czyli numer kontrahenta w kartotece nadawcy.

W podanym przykładzie można użyć również pole nr 2 czyli skrót, które jest także unikalne.

Pozostałe pola kartoteki zostały wypełnione kolejnymi polami z pliku danych, stosownie do ich znaczenia.

Pierwsze trzy sekcje wyjściowe utworzone podczas importu danych:

```
Sekcja KONTRAHENT
    $klucz = 1
    $skrot = "ABM"
$nazwa= "ABM Sp.Cyw."
$miejscowosc = "LUBLIN"
$ulica = "CZERWONYCH SZTANDARÓW 94"
$nip = "737-080-99-99"

Sekcja KONTRAHENT
    $klucz = 2;
$skrot = "APEX"
```

```
$nazwa = "PPHU APEX"
$miestowosc = "Warszawa"
$ulica = "ARABSKA 38"
$nip = "107-838-09-33"
```

Sekcja KONTRAHENT

```
$klucz = 3
$skrot = "BESTER"
$nazwa = "BESTER"
$miestowosc = „KATOWICE"
$ulica = "KLONOWA 22"
$nip = "737-939-00-23"
```

1.6.2 Sekcja Dokument

Sekcja Dokument jest najbardziej złożoną sekcją importu specjalnego, w szczególnym przypadku może zawierać wszystkie pozostałe rodzaje sekcji wyjściowych jako swoje podsekcje. Ilość i wzajemne powiązanie podsekcji w sekcji Dokument jest zależne od rodzaju dokumentu, który jest importowany.

Przykładowo:

Dla dokumentu prostego, WB oraz RK muszą wystąpić co najmniej dwie podsekcje ZAPIS oraz mogą wystąpić podsekcje Transakcja, Rozliczenie oraz podsekcja dowolnej kartoteki

Dla faktur VAT musi wystąpić co najmniej jedna podsekcja Rejestr, oraz mogą wystąpić podsekcje Zapis, Transakcja, Rozliczenie oraz podsekcja dowolnej kartoteki

Rodzaj wypełnianych pól w sekcji Dokument zależy od typu dokumentu i odpowiada informacjom zawartym w części opisowej na formatkach wprowadzania dokumentów w programie Finanse i Księgowość.

\$Klucz

Dla każdego typu importowanego dokumentu (np. faktura zakup lub sprzedaż, raport kasowy, wyciąg bankowy itp.) musi być nadana oddzielna wartość pola \$klucz. Dzięki temu każdy importowany różny typ dokumentu może zostać połączony z innym typem wzorcowym (lub typem zdefiniowanym przez użytkownika) w programie Finanse i Księgowość. Wartość tego pola może być dowolna, nie przekraczająca 100 znaków. Przykładowo jeśli w importowanych danych pole nr 3 zawiera informacje o rodzaju zakup / sprzedaż, a pole nr 4 wyróżnia faktury i rachunki uproszczone, wówczas pole \$klucz może powstać ze złączenia wartości tych dwóch pól, bez względu na to, jakie wartości (liczbowe czy też tekstowe) mogą przyjmować obydwa pola. Przykładowa linia nadająca wartość polu \$ klucz wyglądałaby następująco:

```
$klucz = CAT([3], [4])
```

Jeśliby proste złączenie pól mogło powodować niejednoznaczności, wówczas można pomiędzy wartości pól wstawić dodatkowy znak, np. ukośnik:

```
$klucz = CAT([3], „/”, [4])
```

Dzięki temu każda kombinacja wartości tych pól (faktura zakup, faktura sprzedaż, rachunek uproszczony zakup, rachunek uproszczony sprzedaż) może zostać przyporządkowana przez osobę obsługującą import do innego typu dokumentu w programie Finanse i Księgowość.

Każda nowa wartość pola \$klucz spowoduje otwarcie okna uzgodnień dokumentów, w którym operator będzie mógł przypisać temu kluczowi właściwy typ dokumentu. Wszystkie następne dokumenty z taką samą wartością pola \$klucz zostaną już automatycznie przypisane do tego wybranego typu dokumentu w FK.

Zapisy

Każdy importowany dokument musi mieć co najmniej dwa, bilansujące się zapisy księgowe (po stronie Wn oraz MA). Zapisy mogą być podane jawnie, jako dwie (lub więcej) podsekcje ZAPIS z odpowiednio wypełnionymi polami. Szczegółowo zasady tworzenia podsekcji Zapis są opisane w dalszej części dokumentu, przy opisie sekcji ZAPIS.

W przypadku Faktur oraz Rachunków uproszczonych zapisy mogą być utworzone automatycznie na podstawie danych z rejestrów VAT. Zasady tworzenia tych zapisów są wówczas analogiczne jak dla przejścia z wypełnionego rejestru VAT do formatki wprowadzania dokumentu w programie Finanse i Księgowość. Tworzone są wówczas trzy zapisy, odpowiednio dla kwoty netto, brutto oraz VAT. Szczegółowo zasady automatycznego tworzenia zapisów na podstawie rejestrów są opisane w dalszej części dokumentu, przy opisie sekcji REJESTR

Rozrachunki

Rozrachunki (nowe transakcje oraz rozliczenia transakcji) należy analizować w dwóch przypadkach:

Dla faktur oraz rachunków uproszczonych dla strony z kontem kontrahenta (dla zakupu strona Ma, dla sprzedaży strona Wn). Dla tego przypadku rozrachunki są związane z dokumentem, a sekcja Transakcja lub Rozliczenie musi być bezpośrednio podsekcją dokumentu. W tym przypadku nie ma znaczenia, czy podsekcje zapisów są tworzone jawnie w szablonie, czy też są generowane automatycznie na podstawie rejestrów VAT.

Dla pozostałych typów dokumentu (w tym faktury i rachunki korygujące) oraz dla drugiej strony faktur oraz rachunków uproszczonych. W tym przypadku rozrachunki są związane z konkretnymi zapisami, a sekcja Transakcja lub Rozliczenie musi być bezpośrednio podsekcją wybranego zapisu (podsekcji Zapis)

Aby móc wypełniać rozrachunki na dokumencie, muszą być spełnione następujące warunki:

Zapis, z którym ma być związany rozrachunek musi mieć niezerową kwotę oraz poprawnie podany numer konta. Konto to musi być kontem rozrachunkowym, lub mieć podpętą na ostatnim poziomie analityki kartotekę kontrahentów, pracowników lub urzędów.

Dla rozrachunków związanych z dokumentem (faktury oraz rachunki) musi być jawnie podana podsekcja Zapis z poprawnym numerem konta kontrahenta, lub też w podsekcji rejestr (gdy zapisy są tworzone automatycznie) podany prawidłowy numer konta kontrahenta w polu **\$kontoBrutto**

W jednej sekcji Zapis lub Dokument (dla faktur i RU) może być utworzona najwyżej jedna podsekcja Transakcja odpowiadająca znacznikowi [X] *Nowa transakcja* na dokumencie lub jedna podsekcja Rozliczenie, odpowiadająca rozliczeniu jednej istniejącej transakcji.

W przypadku rozliczenia transakcji musi istnieć dokument i nowa transakcja, która ma być rozliczona. Jeśli program nie odnajdzie transakcji do rozliczenia, wówczas zastąpi rozliczenie nową transakcją.

Przykład szablonu tworzącego nową transakcję dla faktury VAT, z 20-dniowym terminem płatności oraz ustawowymi odsetkami za zwłokę. Dekrety są tworzone automatycznie na podstawie rejestrów VAT:

```
Section(„”, „Dokument”)
{
    $klucz = CAT([3], [4])
    ....
    MakeSec(„Rejestr”)
{
    .....
    $kontoBrutto = „201-1-K”
}

    MakeSec(„Transakcja”)
{
    $termin = 20
    $ustawowe = 1
}
}
```

W powyższym przykładzie litera **K** w numerze konta zostanie zastąpiona właściwym numerem kontrahenta podanego w podsekcji Kontrahent lub wpisanego w polu \$kontrahent w sekcji dokumentu. Zasady tworzenia numerów kont z kartoteką są opisane w dalszej części dokumentu.

1.6.3 Sekcja Zapis

Sekcja Zapis odpowiada jednemu zapisowi księgowemu, czyli jednej kwocie na jedno konto po jednej stronie konta. Sekcja Zapis może wystąpić wyłącznie jako podsekcja sekcji Dokument (nie może wystąpić samodzielnie). Sekcja Zapis nie posiada i nie używa pola \$klucz.

Pojęcia podstawowe dotyczące zapisów:

Zapis - jest to elementarny zapis na jedno konto po jednej stronie. Obowiązującą zasadą jest to, że pierwszy zapis jest zawsze po stronie Wn, a drugi zapis po stronie Ma. Jeśli w jednej pozycji jest więcej niż dwa zapisy, wówczas wszystkie następne zapisy tej pozycji muszą być po tej samej stronie. Odpowiada to zasadom tworzenia zapisów podczas wprowadzania dokumentów w programie Finanse i Księgowość.

Pozycja - jest to elementarna bilansująca się grupa składająca się z dwóch lub więcej zapisów. Jeśli pozycja składa się z więcej niż dwóch zapisów, wówczas po jednej ze stron (wn lub ma) występuje dokładnie jeden zapis, a po stronie przeciwnej pozostałe. Pozycje na dokumencie są numerowane od 0 (zera). Odpowiadają one numerowi pozycji widocznemu w oknie wprowadzania dokumentów, ale przesuniętemu o 1 (na dokumencie numeracja jest od 1). Kolejne pozycje muszą mieć nadane kolejne numery, czyli 0,1,2,3 itd.

Rozbicie - jest to numer kolejny zapisu w ramach pozycji. W każdej pozycji kolejne zapisy tej pozycji są numerowane od 0 do liczby zapisów w pozycji minus jeden. Zasada nadawania numeru rozbicia jest następująca:

0 – pierwszy zapis w pozycji, zawsze po stronie Wn

1 - drugi zapis w pozycji, zawsze po stronie Ma

2,3 itd. Kolejne zapisy w pozycji (o ile pozycja ma więcej niż 2 zapisy). Wszystkie kolejne zapisy muszą być po tej samej stronie (wszystkie po Wn lub wszystkie po Ma).

Numery pozycji i rozbicia są polami w podsekcji zapisu. Prawidłowe numery pozycji i rozbicia muszą być nadane zgodnie z podanymi wyżej zasadami.

UWAGA: Pole \$rozbicie może zostać pominięte (nie wypełnione), ale nadal kolejność poszczególnych sekcji Zapis musi być zgodna z wyżej podanymi zasadami. Pole \$pozycja musi być wypełnione zawsze, jeśli w dokumencie jest więcej niż jedna pozycja.

UWAGA: W sekcji dokumentu podsekcje zapisów (minimum dwie) umieszczane są w ściśle określonej kolejności:

- pierwsza zawsze po stronie WN
- druga zawsze po stronie MA

jeśli zapisy księgowe są rozbite po jednej ze stron na zapisy cząstkowe (np. dla faktury sprzedaży kwota brutto po stronie MA jest rozbita po stronie WN na kwoty netto oraz VAT), wówczas w sekcji dokumentu po dwóch podstawowych sekcjach zapisów, umieszczane są kolejne sekcje rozbicia po odpowiedniej stronie.

Przykłady numeracji pozycji i rozbicia dla różnych schematów księgowania i odpowiadające im podsekcje Zapis w dokumencie.

P - oznacza numer pozycji

R - oznacza numer rozbicia

Pojedyncza pozycja

Strona WN				Strona MA			
P	R	Kwota	Konto	P	R	Kwota	Konto
0	0	345,23	100	0	1	345,23	201-1

Sekcja ZAPIS

\$strona=0 //WN

\$kwota=345,23

\$konto=100

\$pozycja=0

\$rozbicie=0

Sekcja ZAPIS

\$strona=1 //MA

\$kwota=345,23

\$konto=201-1

\$pozycja=0

\$rozbicie=1

Pojedyncza pozycja z rozbiem kwoty po stronie Ma

Strona WN				Strona MA			
-----------	--	--	--	-----------	--	--	--

P	R	Kwota	Konto	P	R	Kwota	Konto
0	0	345,23	100	0	1	200,00	010-1
				0	2	40,00	010-2
				0	3	105,23	010-3

Sekcja ZAPIS

```
$strona=0 //WN
$kwota=345,23
$konto=100
$pozycja=0
$rozbicie=0
```

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=200
$konto=010-1
$pozycja=0
$rozbicie=1
```

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=40,00
$konto=010-2
$pozycja=0
$rozbicie=2
```

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=105,23
$konto=010-3
$pozycja=0
$rozbicie=3
```

Pojedyncza pozycja z rozbiem kwoty po stronie Wn

Strona WN				Strona MA			
P	R	Kwota	Konto	P	R	Kwota	Konto
0	0	200,00	010-1	0	1	345,23	100
0	2	40,00	010-2				
0	3	105,23	010-3				

Sekcja ZAPIS

```
$strona=0 //WN
$kwota=100,00
$konto=010-1
$pozycja=0
$rozbicie=0
```

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=345,23
$konto=100
$pozycja=0
```

\$rozbicie=1

Sekcja ZAPIS

\$strona=0 //WN

\$kwota=40,00

\$konto=010-2

\$pozycja=0

\$rozbicie=2

Sekcja ZAPIS

\$strona=0 //WN

\$kwota=105,23

\$konto=010-3

\$pozycja=0

\$rozbicie=3

Trzy pozycje różnych typów

Strona WN				Strona MA			
P	R	Kwota	Konto	P	R	Kwota	Konto
0	0	345,23	100	0	1	200,00	010-1
				0	2	143,32	010-2
1	0	84,12	100	1	1	84,12	201-1
2	0	3800,00	010-1	2	1	3865,00	100
2	2	65,00	010-2				

Sekcja ZAPIS

\$strona=0 //WN

\$kwota=345,23

\$konto=100

\$pozycja=0

\$rozbicie=0

Sekcja ZAPIS

\$strona=1 //MA

\$kwota=200

\$konto=010-1

\$pozycja=0

\$rozbicie=1

Sekcja ZAPIS

\$strona=1 //MA

\$kwota=143,23

\$konto=010-2

\$pozycja=0

\$rozbicie=2

Sekcja ZAPIS

\$strona=0 //WN

\$kwota=84,12

\$konto=100

\$pozycja=1

\$rozbicie=0

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=84,12
$konto=201-1
$pozycja=1
$rozbicie=1
```

Sekcja ZAPIS

```
$strona=0 //WN
$kwota=3800,00
$konto=010-1
$pozycja=2
$rozbicie=0
```

Sekcja ZAPIS

```
$strona=1 //MA
$kwota=3865,00
$konto=100
$pozycja=2
$rozbicie=1
```

Sekcja ZAPIS

```
$strona=0 //WN
$kwota=65,00
$konto=010-2
$pozycja=2
$rozbicie=2
```

Numer konta może zostać podany bezpośrednio, np. „100”, „321-1” itp. Jednakże w przypadku konta zawierającego kartotekę na ostatnim poziomie analityki nie można podać pełnego numeru konta, gdyż na etapie importu nie jest znany numer np. kontrahenta w kartotece programu Finanse i Księgowość. W takim wypadku należy zastosować specjalną notację, tzn. na ostatnim poziomie analityki zamiast numeru konta należy podać literę oznaczającą typ kartoteki, a następnie podać wartość pola \$klucz użytego przy uzgadnianiu pozycji kartoteki.

Przykłady:

```
201-1-K12
234 -PJanKowalski
220-3-Uzus2
```

Gdzie:

K – kartoteka kontrahentów

P – kartoteka pracowników

U – kartoteka urzędów

A tekst następujący po literze określającej kartotekę jest kluczem identyfikującym pozycję kartoteki u nadawcy.

Jeśli pozycja kartoteki jest podsekcją dokumentu, wówczas najprostszym rozwiązaniem jest zapamiętanie wartości pola \$klucz w podsekcji kartoteki i użycie jej w podsekcji zapisu.

Przykład:

```
Section(„”, „Dokument”)
{
    ...
    MakeSec(„kontrahent”)
    {
        $klucz = [12]
        @@kluczKth=$klucz
    }
}
```

```

.....
}

MakeSec („Zapis”)
{
.....
$konto = CAT („201-1-K”, @@kluczKth)
}
}

```

W powyższym przykładzie założono, że pole nr 12 w danych wejściowych zawiera pole użyte jako klucz kontrahenta (np. numer w kartotece nadawcy). Numer ten został zapamiętany w zmiennej globalnej o nazwie @@kluczKth, a następnie użyty przy stworzeniu numeru konta w zapisie

W przypadku faktur i rachunków uproszczonych VAT, w przypadku gdy dane kontrahenta są umieszczone w podsekcji Kontrahent, wówczas w numerze konta można użyć samej litery **K** bez podawania klucza kontrahenta, np. **201-1-K**. Program automatycznie użyje zamiast litery K właściwego kontrahenta na podstawie danych z sekcji Kontrahent.

Księgowania równoległe automatyczne

W importowanych dokumentach mogą być stosowane księgowania równoległe na zasadach obowiązujących podczas ręcznego wprowadzania dokumentów w programie FK.

Księgowania równoległe mogą być zbiorcze (jedno księgowanie równoległe na sumę wszystkich „4” z księgowania podstawowych) lub przywiązane do poszczególnych pozycji. Na jednym dokumencie może być użyty tylko jeden sposób wypełniania księgowania równoległych. W każdym przypadku zasady numerowania pozycji oraz rozbić dla księgowania równoległych są identyczne jak dla księgowania podstawowych.

Księgowania równoległe zbiorcze muszą być ostatnią pozycją na dokumencie. Dla księgowania równoległych po każdej pozycji zasada jest taka, że pozycja księgowania równoległego występuje bezpośrednio po księgowaniu podstawowym na konto zespołu „4”. Dla wszystkich typów dokumentów (oprócz raportu kasowego) konta zespołu „4” powodujące postanie księgowania równoległych mogą wystąpić tylko po jednej stronie kont (wn lub Ma).

Każdy zapis, który generuje księgowanie równoległe (zapis na konto zespołu „4”) musi mieć ustawiony odpowiedni znacznik, to znaczy w polu sekcji Zapis musi być podstawiona wartość (suma bitów o wartości 4 i 2):

```
$ ZapisRownolegly = 6
```

Również każdy zapis z pozycji księgowania równoległego musi mieć ustawiony znacznik, to znaczy w polu sekcji Zapis musi być podstawiona wartość:

```
$ ZapisRownolegly = 1
```

Dla dokumentu typu Raport Kasowy (RK) księgowania równoległe automatyczne mogą być równocześnie dla strony Wn oraz dla strony Ma. W takim przypadku księgowania równoległe zbiorcze tworzą dwie pozycje: jedną dla sumy kwot na konta zespołu „4” po stronie Wn oraz dla sumy kwot po stronie „Ma”. Kolejność pozycji księgowania równoległych (dla strony Wn oraz Ma) nie ma znaczenia. Dla raportu kasowego znaczniki w sekcjach Zapis księgowania równoległych muszą być następujące:

Dla księgowania równoległych strony Wn:

```
$ ZapisRownolegly = 9
```

Dla księgowania równoległych strony Ma:

```
$ ZapisRownolegly = 17
```

Księgowania równoległe na życzenie

W importowanych dokumentach mogą być stosowane księgowania równoległe na życzenie, na zasadach obowiązujących podczas ręcznego wprowadzania dokumentów w programie FK.

Zasady numerowania pozycji oraz rozbić dla księgowania równoległych są identyczne jak dla księgowania podstawowych. Każdy zapis, który jest związany z księgowaniem równoległym na życzenie (podczas wprowadzania ręcznego jest to bieżący zapis, w momencie tworzenia księgowania równoległego na życzenie) musi mieć ustawiony odpowiedni znacznik, to znaczy w polu sekcji Zapis musi być podstawiona wartość:

```
$ ZapisRownolegly = 2
```

Również każdy zapis z pozycji księgowania równoległego na życzenie musi mieć ustawiony znacznik, to znaczy w polu sekcji Zapis musi być podstawiona wartość:

```
$ ZapisRownolegly = 33
```

1.6.4 Sekcja Rejestr

Sekcja rejestru VAT tworzona jest tylko dla dokumentów mogących posiadać rejestry VAT w programie Finanse i Księgowość. Wyjątkiem jest raport kasowy, który w programie FK może posiadać rejestry VAT, ale nie mogą być one tworzone podczas importu specjalnego. Sekcja Rejestr może być użyta wyłącznie jako podsekcja sekcji Dokument (nie może być samodzielną sekcją). W jednym dokumencie może być jedna lub więcej podsekcji Rejestr. W zależności od sytuacji jedna sekcja Rejestr jest równoważna jednemu rejestrowi VAT na dokumencie, jednemu typowi zakupu (A,B lub C) lub jednej lub więcej stawek w wybranym rejestrze VAT.

Pole \$Klucz

Każda sekcja Rejestr musi mieć prawidłowo wypełnione pole \$klucz.

Dla każdego typu importowanego dokumentu posiadającego rejestr VAT oraz dla każdego rodzaju rejestru (zwykły, środków trwałych, itp.) musi być nadana oddzielna wartość pola \$klucz. Dzięki temu każdy oddzielny rodzaj rejestru VAT w każdym importowanym dokumencie może zostać połączony z innym rejestrem VAT w programie Finanse i Księgowość. Wartość tego pola może być dowolna.

Każda wartość pola \$klucz jest związana dokładnie z jedną definicją rejestru VAT w programie Finanse i Księgowość, niezależnie z jakim typem dokumentu ten rejestr jest związany. Przykładowo jeśli dla wartości pola \$klucz równej „RVS” zostanie użyty rejestr faktur VAT sprzedaży, to ten właśnie rejestr zostanie użyty dla każdego kolejnego wystąpienia tej wartości pola \$klucz, nawet jeśli będzie to np. dla dokumentu typu rachunek uproszczony sprzedaż. Oczywiście w takim przypadku procedury importu specjalnego zgłoszą błąd nieprawidłowego przypisania rejestru do dokumentu. Aby uniknąć tego typu błędów w polu \$klucz musi być unikalna wartość dla każdej kombinacji typu dokumentu i rodzaju rejestru VAT, czyli do informacji identyfikującej rodzaj rejestru (zwykły, środków trwałych itp.) należy dodać informacje jednoznacznie identyfikujące typ dokumentu (faktura, rachunek, zakup, sprzedaż). Najprościej dodać do klucza rejestru wartość klucza dokumentu.

Przykład:

```
Section(„”, „Dokument”)
{
    $klucz = CAT([3], [4])
    ....
    MakeSec(„Rejestr”)
    {
        $klucz = CAT( $$klucz, [9])
        ....
    }
}
```

W powyższym przykładzie założono, że pole nr 3 z w pliku danych określa rodzaj faktura / rachunek, pole nr 4 określa zakup / sprzedaż, a pole nr 9 określa rodzaj rejestru VAT (zwykły, środków trwałych, towarów, usług itp.). Pole \$klucz w sekcji dokumentu utworzono ze złączenia wartości pól 3 oraz 4, a klucz sekcji Rejestr ze złączenia pola \$ klucz z sekcji dokumentu (dwa znaki \$\$ oznaczają pole z głównej sekcji – w tym przypadku pole \$klucz z sekcji dokumentu) oraz pola 9 wyróżniającego typ rejestru VAT. Nie ma tu znaczenia jakie wartości (liczbowe czy też tekstowe) mogą przyjmować użyte pola, jeśli tylko łączna ich długość nie przekracza 100 znaków.

Każda nowa wartość pola \$klucz spowoduje otwarcie okna uzgodnień rejestrów VAT, w którym operator będzie mógł przypisać temu kluczowi właściwy rejestr. Wszystkie następne rejestry z taką samą wartością pola \$klucz zostaną już automatycznie przypisane do tego wybranego rejestru w programie FK.

Zasady łączenia sekcji Rejestr

Podczas importu specjalnego można utworzyć kilka sekcji Rejestr, które następnie zostaną połączone w jeden rejestr VAT w importowanym dokumencie. Zasada jest taka, że w jeden rejestr VAT zostaną połączone wszystkie podsekcje Rejestr z jednego dokumentu, które mają taką samą wartość pola \$klucz.

Dzięki temu mechanizmowi mogą być połączone różne rodzaje zakupy (A,B,C), można również, jeśli jest taka potrzeba, dla każdej oddzielnej stawki VAT stworzyć oddzielną sekcję Rejestr, a wszystkie sekcje zostaną połączone w jeden rejestr z zachowaniem przypisania do odpowiednich stawek procentowych VAT.

Określenie rodzaju zakupów A,B,C

Dla każdego dla każdego rodzaju zakupu (A-związanego ze sprzedażą opodatkowaną, B – zw. Ze sprz. zwolnioną i C – zw. Ze sprz. opodatkowaną i zwolnioną) należy utworzyć oddzielną sekcję Rejestr.

Dla każdej z tych sekcji należy prawidłowo wypełnić pola \$ABC. Odpowiednie wartości są następujące:

1 – Zakupy A, związane ze sprzedażą opodatkowaną

2 – Zakupy B, związane ze sprzedażą zwolnioną

4 – Zakupy C, związane ze sprzedażą opodatkowaną i zwolnioną

Jeśli w jednym dokumencie występuje kilka rodzajów zakupu, muszą być one umieszczone w oddzielnych sekcjach Rejestr. Jeśli Wszystkie te rodzaje zakupu mają być umieszczone docelowo w tym samym rejestrze VAT, wówczas wszystkie sekcje Rejestr muszą mieć identyczną wartość pola \$klucz. Jeśli dla każdego rodzaju zakupu zdefiniowano w programie FK osobny rejestr VAT, wówczas każda sekcja Rejestr musi mieć inną wartość pola \$klucz (np. dodana do pola \$klucz informacja o rodzaju A,B,C). Wówczas podczas importu każdy rodzaj zakupu będzie mógł być przypisany do innego rejestru VAT.

Pole \$abc może być wypełnione wyłącznie dla rejestrów zakupu. Dla rejestrów sprzedaży pole to powinno być puste lub mieć wartość zero.

Określenie stawek procentowych VAT

W jednej sekcji Rejestr może być zawarta jedna stawka procentowa VAT, co jest istotną różnicą w stosunku do pierwotnych implementacji importu obowiązujących dla formatu AM, gdzie w jednej sekcji Rejestr mogły być podane maksymalnie 4 stawki niezerowe oraz stawka 0% i Zwolnione. Ten wariant przekazywania danych nie jest zalecany od wersji 5.35, bowiem nie można w nim zawrzeć bardziej nietypowych szczegółów rejestru wprowadzonych do programu wraz ze zmieniającymi się przepisami prawa.

Tworzenie zapisów na podstawie rejestrów

Import specjalny ma możliwość automatycznego tworzenia zapisów dokumentu na podstawie danych z rejestrów VAT. Działa tu mechanizm analogiczny, jak podczas wprowadzania dokumentów w programie FK, przy przejściu ze strony rejestru na stronę dokumentu tworzone są automatycznie zapisy. Również podczas importu specjalnego są tworzone automatycznie podsekcje zapisów, oczywiście tylko wówczas, gdy nie zostaną one utworzone w sposób jawny w szablonie. Automatycznie tworzone są zapisy po jednej stronie dla kwoty netto i VAT oraz po drugiej stronie dla kwoty brutto. Możliwe jest podanie kont, jakie mają być użyte w poszczególnych zapisach. W tym celu należy wypełnić odpowiednie pola sekcji Rejestr:

\$kontoNetto - konto dla kwoty netto

\$kontoVat - konto dla kwoty VAT

\$kontoBrutto - konto dla kwoty brutto

Dla dokumentów zakupu kwota brutto księgowana jest po stronie Ma, a kwota netto i VAT po stronie Winien. Dla dokumentów sprzedaży jest odwrotnie.

Jeśli konto dla kwoty brutto jest kontem z kartoteką kontrahentów, wówczas w numerze konta na ostatnim poziomie analityki można podać samą literę **K**, która zostanie zamieniona na numer kontrahenta, zgodnego z danymi umieszczonymi w podsekcji Kontrahent w dokumencie importowanym.

Przykład:

```
Section(„”, „Dokument”)
{
    $klucz = CAT([3], [4])
    ....
    MakeSec(„Kontrahent”)
    {
        $klucz =[8]
    ....
    }
    MakeSec(„Rejestr”)
    {
        ....
        $kontoBrutto = „201-1-K”
    }
}
```

```
}  
  
}
```

Jeśli nie chcemy tworzyć zapisów automatycznie, wówczas należy utworzyć podsekcję Zapis w sposób jawny, jednak należy zachować zasady tworzenia zapisów opisane wcześniej, oraz zachować sposób tworzenia zapisów odpowiadający dokumentom wprowadzanym ręcznie do programu Finanse i Księgowość.

Jeśli w importowanym dokumencie chcemy umieścić dodatkowo księgowania równoległe, wówczas wszystkie zapisy (podstawowe i równoległe) muszą zostać jawnie utworzone w szablonie.

Wielowalutowość

Dla dokumentów typu faktura walutowa, w sekcji *Rejestr* dodano trzy nowe pola. Pola te zaimplementowano ze względu na przesyłanie dodatkowych informacji przez program Handel. Dotyczą one wartości kwot netto, brutto i vat reprezentowanych w walucie. Dla pozostałych typów dokumentów oraz podczas importu danych z innych źródeł, wartości tych pól nie są przesyłane. Wartości nowych pól prezentowane są (dla faktury walutowej) w oknie edycji rejestrów vat.

Dodatkowe pola sekcji *Rejestr* dla faktur walutowych:

NettoWaluta

Kwota netto rejestru w walucie.

```
$nettoWaluta =2991.56
```

BruttoWaluta

Kwota brutto rejestru w walucie.

```
$bruttoWaluta =3078.95
```

VatWaluta

Kwota podatku VAT w walucie (jeżeli pole stawka ma wartość większą od zera).

```
$vatWaluta =87.39
```

Przykład sekcji wyjściowej rejestru dla faktury walutowej

```
Rejestr{  
    skrot =rZPV  
    Nazwa =Zakup VAT  
    Rodzaj =0  
    ABC =1  
    datarej =2010-07-15  
    okres =2010-07-01  
    stawka =3  
    brutto =5.85  
    netto =5.68  
    vat =0.17  
    bruttoWaluta =3078.95  
    nettoWaluta =2991.56  
    vatWaluta =87.39  
}
```

O tym czy dokument jest nowego typu informuje nas pole *obsługuj jak* znajdujące się w sekcji *Dokument*.

1.6.5 Sekcja Pozycja Vat

Sekcja Pozycja Vat może być użyta jedynie jako podsekcja sekcji Rejestr (nie może być samodzielną sekcją). W związku z powyższym jest ona tworzona jedynie dla dokumentów mogących posiadać rejestr vat w programie Finanse i Księgowość. Dla jednego rejestru może zostać utworzona jedna lub więcej sekcji Pozycja Vat. Sekcja może reprezentować pozycje rejestru vat będące usługą lub towarem. W sekcji Pozycja Vat nie występuje i nie jest używane pole \$klucz.

Pola sekcji Pozycja Vat:

Stawka

Stawka vat dla pozycji rejestru. Jeżeli przyjmuje wartości dodatnie lub zero, interpretowane są one jako wartości procentowe stawki vat, np.:

```
$stawka = 22
```

odpowiada stawce vat 22%.

Pole może przyjmować wartości ujemne z zakresu od -3 do -1. Wartości te interpretowane są w specjalny sposób, gdzie:

- wartość -1 interpretowana jest jako stawka „zwolnione”
- wartość -2 interpretowana jest jako stawka „nie opodatkowane”
- wartość -3 interpretowana jest jako stawka „bez udziału”

Wszystkie pozostałe wartości ujemne interpretowane są jako „nieznana stawka”

```
$stawka = -1
```

Stawka_pl

Reprezentuje stawkę vat jaka jest naliczana w Polsce dla pozycji rejestru. Wartość pola Stawka_pl może być różna od wartości przypisanej polu Stawka. Zakres przyjmowanych przez opisywane pole wartości oraz ich interpretacja jest zgodna z opisem dla pola Stawka.

Przykład:

```
$stawka_pl = 22
```

Opis

Nazwa pozycji vat. Może zawierać maksymalnie 200 znaków (pozostałe znaki są obcinane), np.:

```
$opis = owoce i warzywa
```

Wartość

Wartość netto pozycji vat w złotych, np.:

```
$wartosc = 11589.03
```

Usługa

Pole przyjmuje wartości 0 lub 1. Określa czy operacja gospodarcza reprezentowana przez pozycję vat dotyczy towarów (0) czy usług (1), np.:

```
$usluga = 1
```

Ue

Pole przyjmuje wartości 0 lub 1. Określa czy operacja gospodarcza reprezentowana przez pozycję vat jest realizowana na terytorium Unii Europejskiej (1), czy poza nią (0), np.:

```
$ue = 1
```

Podczas importu specjalnego, odczytane z pliku tekstowego pozycje vat są, dołączane do właściwych rejestrów vat. Struktura pliku tekstowego importu specjalnego decyduje, poprzez zagnieżdżanie sekcji Pozycja Vat wewnątrz sekcji rejestr, jakie pozycje vat należy dołączyć do określonego rejestru.

W przypadku niezgodności atrybutów UE lub Usługa pomiędzy rejestrem a przypisaną do niego pozycją vat konieczne jest odpowiednie przeprocesowanie takiego rejestru. Procedura procesowania rejestrów vat jest wykonywana podczas importu specjalnego. W wyniku jej działania, mogą powstać nowe encje rejestru vat z dołączonymi do nich pozycjami vat.

Opis procedury procesowania rejestrów vat

Procedura procesowania rejestrów vat w ściśle określonych przypadkach (przedstawionych poniżej) ustawia wstępnie wartość znacznika Ue dla importowanych rejestrów.

Dla dokumentów eksportowych atrybut Ue jest ustawiany na 0 dla wszystkich rejestrów vat.

Dla dokumentów zakupu, dla rejestrów typu Faktura Wewnętrzna WNT atrybut Ue ustawiany jest zawsze na 1. Dodatkowo w przypadku stwierdzenia identycznych rejestrów (różniących się jedynie wartością atrybutu

Ue) w ramach jednego dokumentu, kwoty netto, brutto i vat, takich rejestrów są sumowane. Zdublowane rejestry są usuwane, a przypisane do nich usługi są dołączane do rejestru z sumarycznymi kwotami. Dla dokumentów typu Wewnątrzwspólnotowa Dostawa Towaru, dla wszystkich rejestrów, wartość atrybutu Ue ustawiana jest na 1.

Dla dokumentów typu Wewnątrzwspólnotowe Nabycie Towaru oraz Wewnątrzwspólnotowa Dostawa Towaru, pochodzących z programu Handel (Symfonia, Forte) w wersji starszej niż 2010.1, zmieniana jest wartość atrybutu Ue rejestrów vat na 1. Dla dokumentów typu WDT istnieje dodatkowy warunek: stawka vat rejestru musi wynosić 0% lub Nieopodatkowane. W tym celu czytane jest pole wersjaWewn (Forte) lub Wersja_programu (Symfonia) sekcji INFO. Jeżeli wartości powyższych pól są odpowiednio mniejsze od 308 i 86, algorytm procesowania rejestrów vat przyjmuje, że dokumenty pochodzą z wersji starszej niż 2010.1 programu Handel.

Ustawione wstępnie, na wyżej opisanych zasadach, wartości atrybutu Ue mogą ulec zmianie w wyniku późniejszego procesowania rejestrów vat ze względu na dołączone do nich pozycje vat.

W procedurze procesowania rejestrów można wyróżnić, ze względu na różne dane początkowe, kilka przypadków, różniących się sposobem przetwarzania danych.

Przypadek 1

Jeżeli atrybuty Ue i Usługa mają wartości zgodne dla rejestru i pozycji vat, a wartość pozycji vat jest mniejsza bądź równa kwocie netto rejestru, pozycja vat jest dołączana do rejestru bez zmiany jego atrybutów.

Przypadek 2

Jeżeli zostanie stwierdzona niezgodność wartości atrybutu Ue i/lub Usługa pomiędzy rejestrem, a przypisaną do niego pozycją vat, oraz kiedy kwota pozycji vat jest mniejsza od kwoty netto rejestru, powstaje nowe wystąpienie rejestru vat. Atrybuty Ue i Usługa nowej encji rejestru są ustawiane zgodnie z wartościami w Pozycji Vat, która spowodowała rozbitcie. Kwota netto encji rejestru odpowiada kwocie z pola Wartość Pozycji Vat, kwoty brutto i vat są ponownie obliczane. Do nowego wystąpienia rejestru dołączana jest pozycja vat, która spowodowała rozbitcie.

Kwota rozbijanego rejestru zostaje zmniejszona o kwotę przypisaną do nowej encji rejestru vat, a kwoty brutto i vat są ponownie przeliczone. Wartości atrybutów Ue i Usługa rozbijanego rejestru nie ulegają zmianie.

Przykład:

Dane wejściowe z pliku tekstowego.

Rejestr

```
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28
    okres =2009-05-01
    stawka =0
    brutto =301.10
    netto =301.10
    vat =0
    uslug =0
    ue =0
    Pozycja VAT
    {
        Stawka =0
        Stawka_pl =22
        Opis =Usługa przewozowa
        Wartość =100.00
    }
```

```

    rejestrVat =sprzedaż VAT
    Usługa =1
    UE =0
}
Pozycja VAT
{
    Stawka =0
    Stawka_pl =22
    Opis =Usługa specjalna
    Wartość =80.00
    rejestrVat =sprzedaż VAT
    Usługa =1
    UE =1
}
}

```

W wyniku procesowania rejestrów otrzymujemy:

```

Rejestr //nowa encja rejestru w wyniku rozbicia pozycją vat
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28
    okres =2009-05-01
    stawka =0
    brutto =100.00
    netto =100.00
    vat =0
    usługa =1
    ue =0
    Pozycja VAT
    {
        Stawka =0
        Stawka_pl =22
        Opis =Usługa przewozowa
        Wartość =100.00
        rejestrVat =sprzedaż VAT
        Usługa =1
        UE =0
    }
}
Rejestr //nowa encja rejestru w wyniku rozbicia pozycją vat
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28
    okres =2009-05-01
    stawka =0

```

```

    brutto =80.00
    netto =80.00
    vat =0
    usługa =1
    ue =1

Pozycja VAT
{
    Stawka =0
    Stawka_pl =22
    Opis =Usługa specjalna
    Wartość =80.00
    rejestrVat =sprzedaż VAT
    Usługa =1
    UE =1
}
}
Rejestr
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28
    okres =2009-05-01
    stawka =0
    brutto =121.10
    netto =121.10
    vat =0
    usługa =0
    ue =0
}

```

Omówienie przykładu:

Dwie pozycje vat dołączone do rejestru, w wyniku niezgodności parametrów Ue i/lub Usługa, spowodowały jego „rozbicie”, na dwie nowe encje. Ponieważ wartość pozycji vat nie wyczerpuje kwoty rejestru, w wyniku procesowania powstaje rejestr bez dołączonych usług. Jego kwota netto jest równa różnicy pomiędzy pierwotną kwotą rejestru, a sumą wartości Pozycji Vat. W związku z tym, że rejestr ten nie posiada dołączonych usług, jego atrybuty Ue i Usługa nie ulegają modyfikacji.

Przypadek 3

Jeżeli zostanie stwierdzona niezgodność wartości atrybutu Ue i/lub Usługa pomiędzy rejestrem, a przypisaną do niego pozycją vat, kiedy kwota netto rejestru jest równa kwocie pozycji vat, nie powstaje nowa encja rejestru. W istniejącym rejestrze ustawiane są jedynie atrybuty Ue i/lub Usługa w taki sposób aby były zgodne z dołączoną do niego pozycją vat.

Przykład:

Dane wejściowe z pliku tekstowego

```

Rejestr
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28

```

```

    okres =2009-05-01
    stawka =0
    brutto =200
    netto =200
    vat =0
    usluga =0
    ue =0
Pozycja VAT
{
    Stawka =0
    Stawka_pl =22
    Opis =Usługa przewozowa
    Wartość =200
    rejestrVat =sprzedaż VAT
    Usługa =1
    UE =0
}
}

```

W wyniku procesowania rejestrów vat otrzymujemy:

```

Rejestr
{
    Skrot =rSPV
    Nazwa =Sprzedaż VAT
    Rodzaj =1
    datarej =2009-05-28
    okres =2009-05-01
    stawka =0
    brutto =200
    netto =200
    vat =0
    usluga =1
    ue =0
Pozycja VAT
{
    Stawka =0
    Stawka_pl =22
    Opis =Usługa przewozowa
    Wartość =200
    rejestrVat =sprzedaż VAT
    Usługa =1
    UE =0
}
}

```

Nie stwierdzono przypadku biznesowego, uzasadniającego sytuację w której wartość Pozycji Vat byłaby większa niż kwota netto rejestru do którego pozycja ta jest dołączana. Zaimplementowany algorytm obsługuje jednak przyjmowanie takich danych początkowych poprzez zwiększenie kwoty netto rejestru, tak aby była równa wartości pozycji. Znaczniki Ue i Usługa rejestru ustawiane są zgodnie z wartościami przypisanej do rejestru Pozycji Vat. Utworzona zostanie również nową encja rejestru vat z ujemną kwotą,

równą różnicę pomiędzy kwotą netto rejestru a wartością Pozycji Vat. Dzięki utworzeniu takiej encji suma rejestrów vat nie ulega zmianie i jest zgodna z kwotą dokumentu.

W związku z prezentowanymi powyżej przykładami należy podkreślić, że program Handel (Symfonia i Forte) w obecnej wersji (2010.1) nie przesyła informacji o wartościach pól „usługa” i „ue” dla rejestru vat. Pola te zostały umieszczone w przykładach jedynie w celu zwiększenia ich czytelności, a ich wartość jest równa wartości domyślnej tych pól (0). Takie wartości, wobec braku tych pól, przyjmuje do analizy, jako wartości początkowe, algorytm procesowania rejestrów vat. W przypadku, kiedy program stwierdzi obecność powyższych pól w importowanym pliku tekstowym (np. w wyniku ręcznego dopisania przez użytkownika) do analizy i procesowania przyjmowane są bieżące wartości.

1.6.6 Sekcja Transakcja

Sekcja Transakcja odpowiada znacznikowi [x] *nowa transakcja* w rozrachunkach, podczas wprowadzania dokumentów w programie FK. Aby prawidłowo wypełniać nowe transakcje należy pamiętać, że transakcja może być związana albo z pojedynczym zapisem, albo z całym dokumentem:

Dla faktur oraz rachunków uproszczonych dla strony kwoty brutto (dla zakupu strona Ma, dla sprzedaży strona Wn) transakcje są związane z dokumentem, a sekcja Transakcja musi być bezpośrednio podsekcją dokumentu. W tym przypadku nie ma znaczenia, czy podsekcje zapisów są tworzone jawnie w szablonie, czy też są generowane automatycznie na podstawie rejestrów VAT

Dla pozostałych typów dokumentu (w tym faktury i rachunki korygujące) oraz dla drugiej strony faktur oraz rachunków uproszczonych transakcje są związane z konkretnymi zapisami, a sekcja Transakcja musi być bezpośrednio podsekcją wybranego zapisu (podsekcji Zapis) lub podsekcją dokumentu, ale z poprawnie wypełnioną wartością pola **idDlaRozliczen**. Zasady wypełniania tego pola opisane są w punkcie **Zasady powiązanie rozrachunków z zapisami**.

Sekcja transakcja może być użyta wyłącznie jako podsekcja sekcji Dokument lub Zapis, nie może występować samodzielnie. W sekcji Transakcja nie występuje i nie jest używane pole \$klucz.

Termin płatności

Termin płatności dla nowej transakcji może być podany na dwa sposoby:

Jako pełna data terminu płatności, np.

```
$termin = „1999-04-12”
```

Jako liczba dni licząc od daty dokumentu, czyli od pola \$datadok w sekcji

Dokument:

```
$termin = 14
```

Oczywiście termin płatności zwykle jest informacją w pliku danych, wówczas należy zrobić odpowiednie podstawienie nie wartości bezpośredniej, ale wartości pola wejściowego:

```
$termin = Date([15]) //dla daty w formacie pełnym
```

```
$termin = [14] //dla liczby dni licząc od daty dokumentu
```

Odsetki

Dla nowej transakcji należy określić rodzaj odsetek za zwłokę. Można zastosować odsetki ustawowe, zdefiniowane w programie Finanse i Księgowość, lub też podać stałą stopę odsetek.

Dla odsetek ustawowych podajemy:

```
$ustawowe = 1
```

Dla odsetek stałych musimy podać stopę procentową odsetek dziennie:

```
$stopa = 0.015
```

Nie ma możliwości podania stopy odsetek rocznie. Program FK nalicza odsetki za każdy dzień kalendarzowy, bez zwracania uwagi na dni ustawowo wolne od pracy (niedziele i święta).

Zaliczka

Nowa transakcja może być oznaczona jako zaliczka. Odpowiada to polu [x] *zaliczka* podczas wypełniania rozrachunków w programie FK. Aby oznaczyć nową transakcję jako zaliczkę, należy podstawić wartość niezerową do pola \$zaliczka:

```
$zaliczka = 1
```

Brak pola \$zaliczka w sekcji Transakcja, lub wartość zerowa tego pola oznacza zwykłą transakcję.

Kwota

Kwotą nowej transakcji będzie zawsze kwota zapisu, z którą związana jest sekcja Transakcja. W przypadku transakcji związanej z całym dokumentem, kwotą transakcji będzie suma wszystkich zapisów (bez księgowañ równoległych) po stronie odpowiadającej transakcji. Jest to zgodne z zasadami obowiązującymi podczas ręcznego wprowadzania dokumentów do programu FK

Konto transakcji

W sekcji Transakcja nie ma pola, w którym podajemy numer konta nowej transakcji.

Jeśli sekcja transakcja jest podsekcją sekcji Zapis, wówczas numer konta pobierany jest z sekcji Zapis. Aby nowa transakcja w dokumencie została utworzona, konto to musi być poprawne, tzn. musi istnieć, być ostatnim poziomem analityki oraz być kontem rozrachunkowym (lub mieć podpiętą kartotekę kontrahentów, pracowników lub urzędów). Jeśli jest to konto z kartoteką, musi dodatkowo istnieć odpowiednia pozycja w tej kartotece.

Jeśli nowa transakcja jest związana z dokumentem, wówczas w dokumencie odszukiwane są wszystkie zapisy po odpowiedniej stronie (dla zakupu po stronie Ma, dla sprzedaży po Wn), oprócz księgowañ równoległych. Kwota ze wszystkich tych zapisów jest sumowana i tworzy kwotę nowej transakcji. Konto we wszystkich tych zapisach musi być identyczne (analogicznie jak dla ręcznego wprowadzania faktury i RU w programie FK) i spełniać podane wyżej wymagania. Jeśli jest to konto z kartoteką kontrahentów, wówczas numer kontrahenta numerze konta musi się zgadzać z kontrahentem w części opisowej dokumentu (w podsekcji Kontrahent). Jeśli program stwierdzi błąd w numerze konta (nie spełnione powyższe warunki), wówczas zgłosi komunikat:

(611) Nie znaleziono zapisu odpowiadającego sekcji rozrachunków.

i nowa transakcja nie zostanie utworzona.

Przykład nowej transakcji związanej z zapisem:

```
Section(„”, „Dokument”)
{
    ....
    MakeSec(„kontrahent”)
    {
        $klucz = [12]
        @@kth = $klucz//zapamiętujemy klucz kontrahenta
        ....
    }
    MakeSec(„Zapis”)
    {
        $konto = CAT(„201-1-K”, @@kth) //konto kontrahenta
        $strona=1 //Ma
        $kwota=[24]
        .....
        MakeSec(„Transakcja”)
        {
            $termin = 28 //28 dni od daty dokumentu
            $ustawowe=1 //odsetki ustawowe
            $znacznik = 73
        }
    }
}
```

W powyższym przykładzie w sekcji Kontrahent został zapamiętany klucz kontrahenta w zmiennej globalnej @@kth, a następnie użyty w numerze konta w podsekcji Zapis. Nowa transakcja będzie miała odsetki ustawowe, i zostanie opatrzona znacznikiem 'I' oraz 28 dniowym terminem płatności.

1.6.7 Sekcja Rozliczenie

Sekcja Rozliczenie odpowiada rozliczeniu istniejącej transakcji w rozrachunkach, podczas wprowadzania dokumentów w programie FK. Aby prawidłowo wypełniać rozliczenia należy pamiętać, że rozliczenie podobnie jak nowa transakcja może być związana albo z pojedynczym zapisem, albo z całym dokumentem:

Dla faktur oraz rachunków uproszczonych dla strony kwoty brutto (dla zakupu strona Ma, dla sprzedaży strona Wn) rozliczenia są związane z dokumentem, a sekcja Rozliczenie musi być bezpośrednio podsekcją dokumentu. W tym przypadku nie ma znaczenia, czy podsekcje zapisów są tworzone jawnie w szablonie, czy też są generowane automatycznie na podstawie rejestrów VAT

Dla pozostałych typów dokumentu (w tym faktury i rachunki korygujące) oraz dla drugiej strony faktur oraz rachunków uproszczonych rozliczenia są związane z konkretnymi zapisami, a sekcja Rozliczenie musi być bezpośrednio podsekcją wybranego zapisu (podsekcji Zapis) lub podsekcją dokumentu, ale z poprawnie wypełnioną wartością pola **idDlaRozliczen**. Zasady wypełniania tego pola opisane są w punkcie **Zasady powiązanie rozrachunków z zapisami**.

Sekcja transakcja może być użyta wyłącznie jako podsekcja sekcji Dokument lub Zapis, nie może występować samodzielnie. W sekcji Transakcja nie występuje i nie jest używane pole \$klucz.

Konto

W sekcji Rozliczenie nie ma pola, w którym podajemy numer konta rozliczenia.

Jeśli sekcja Rozliczenie jest podsekcją sekcji Zapis, wówczas numer konta pobierany jest z sekcji Zapis.

Jeśli rozliczenie jest związane z dokumentem, wówczas w dokumencie odszukiwane są wszystkie zapisy po odpowiedniej stronie (dla zakupu po stronie Ma, dla sprzedaży po Wn), oprócz księgowiń równoległych. Kwota ze wszystkich tych zapisów jest sumowana i tworzy kwotę rozliczenia. Konto we wszystkich tych zapisach musi być identyczne (analogicznie jak dla ręcznego wprowadzania faktury i RU w programie FK) i spełniać podane wyżej wymagania. Jeśli jest to konto z kartoteką kontrahentów, wówczas numer kontrahenta numerze konta musi się zgadzać z kontrahentem w części opisowej dokumentu (w podsekcji Kontrahent). Jeśli program stwierdzi błąd w numerze konta (nie spełnione powyższe warunki), wówczas zgłosi komunikat:

(611) Nie znaleziono zapisu odpowiadającego sekcji rozrachunków.

i rozliczenie nie zostanie dokonane.

Dokument rozliczany

Aby rozliczenie mogło zostać dokonane, musi istnieć właściwy dokument tworzący nową transakcję, która ma zostać rozliczona. Informacją która jednoznacznie identyfikuje transakcję do rozliczenia jest numer konta transakcji (i rozliczenia) oraz numer własny dokumentu do rozliczenia. Ponieważ jeden kontrahent nie może wystawić dwóch różnych dokumentów (faktur) o tym samym numerze własnym, więc wystarczy to do jednoznacznej identyfikacji transakcji. Numer konta jest określany w sposób opisany wcześniej, natomiast numer dokumentu rozliczanego (tworzącego nową transakcję) musi być podany jawnie w sekcji Rozliczenie. Służy temu pole o nazwie \$Dsymbol.

```
$dsymbol = „FS/04/99/0037”
```

Rozliczenie można utworzyć tylko wówczas, gdy w danych wejściowych znajduje się informacja o numerze własnym rozliczanego dokumentu. Bez tej informacji nie jest możliwe zidentyfikowanie transakcji do rozliczenia. W przypadku wielu terminów płatności transakcji (linia ERP) transakcje są rozliczane zgodnie z kolejnością terminów.

Przykład:

```
Section(„”, „Dokument”)
{
    ....
    MakeSec(„Zapis”)
    {
        $konto = CAT(„201-1-K”, [9])
        $strona=1 //Ma
        $kwota=[24]
        .....
        MakeSec(„Rozliczenie”)
        {
            $Dsymbol = [23]
```

```

    }
}
}

```

W powyższym przykładzie założono, że pole nr 9 zawiera informację wykorzystaną jako klucz podczas importu danych kontrahenta, a pole nr 23 numer własny dokumentu który ma zostać rozliczony.

Kwota rozliczenia

Kwotą rozliczenia będzie zawsze kwota zapisu, z którą związana jest sekcja Rozliczenie. W przypadku rozliczenia związanego z całym dokumentem, kwotą rozliczenia będzie suma wszystkich zapisów (bez księgowości równoległych) po stronie odpowiadającej rozliczeniu. Jest to zgodne z zasadami obowiązującymi podczas ręcznego wprowadzania dokumentów do programu FK

1.6.8 Sekcja AtrybutWaluta

Sekcja *AtrybutWaluta* może być użyta jedynie jako podsekcja sekcji *Dokument* (nie może być samodzielną sekcją). Sekcja wykorzystywana jest do przesyłania kursów VAT i CIT/PIT dla faktury walutowej. Dla pozostałych typów faktur sekcja *AtrybutWaluta* nie jest przesyłana i procesowana w programie Finanse i Księgowość. Dla jednego dokumentu walutowego, który posiada zapisy walutowe (nierównoległe) lub rejestry walutowe, zawsze przesyłane są dwie sekcje *AtrybutWaluta*. Reprezentują one odpowiednio kurs VAT i CIT/PIT. W przeciwnym wypadku taki dokument nie zostanie zwalidowany jako poprawny. Dane przesyłane w tej sekcji prezentowane są (dla faktur walutowych) w oknie edycji dokumentu oraz vat.

W sekcji *AtrybutWaluta* nie występuje i nie jest używane pole \$klucz.

Pola sekcji *AtrybutWaluta*:

Dane

Zawiera informację o rodzaju przesyłanych danych. Może przyjmować jedną z dwóch wartości tekstowych: „kurs VAT” lub „kurs CIT/PIT”. Wypełnienie pola wymagane.

```
$dane = kurs CIT/PIT
```

Waluta

Reprezentuje nazwę waluty (np: trzyznakowy kod ISO). Wypełnienie pola wymagane.

```
$waluta = CSK
```

Kurs

Wartość kursu waluty. Wypełnienie pola wymagane.

```
$kurs =0.001900
```

Opis

Dodatkowy opis użytkownika. Wypełnienie pola opcjonalne.

```
$ opis = Tabela kursów nr 61/A/NBP/2010
```

Data

Data przesyłanego kursu waluty. Wypełnienie pola opcjonalne.

```
$ data = 2010-07-15
```

Podczas importu specjalnego, atrybuty walutowe odczytane z pliku tekstowego (faktura walutowa), dołączane są do właściwych dokumentów. Struktura pliku tekstowego importu specjalnego decyduje, poprzez zagnieżdżanie sekcji *AtrybutWaluta* wewnątrz sekcji *Dokument*, jakie atrybuty walutowe należy dołączyć do określonego dokumentu.

Przykład sekcji wyjściowej

```

Dokument
{
    ...
    AtrybutWaluta{
        kurs =0.001900
    }
}

```

```

    waluta =CSK
    dane =kurs VAT
    opis = Tabela kursów nr 61/A/NBP/2010
    data = 2010-07-15
}
AtrybutWaluta{
    kurs =0.001900
    waluta =CSK
    dane =kurs CIT/PIT
    opis = Tabela kursów nr 64/A/NBP/2010
    data = 2010-07-15
}
...
}

```

1.6.9 Zasady powiązania rozrachunków z zapisami

Sekcje *Transakcja* oraz *Rozliczenie* mogą być powiązane z odpowiednimi sekcjami zapisów bezpośrednio lub pośrednio.

Powiązanie bezpośrednie polega na tym, że właściwa sekcja rozrachunków jest po prostu podsekcją konkretnej sekcji *Zapis*. W takiej sytuacji wszystkie dodatkowe informacje potrzebne do utworzenia rozrachunków pobierane są z sekcji zapisu (np. kwota, konto) w której umieszczona jest podsekcja rozrachunków.

Przykład sekcji wyjściowej:

```

Dokument{
    ...
    Zapis{
        $Kwota=12,89
        $Konto=201-1
        ...
        Transakcja{
            $Termin=14
            ...
        }
    }
}

```

Powiązanie pośrednie polega na tym, że sekcja *Transakcja* lub *Rozliczenie* jest umieszczone jako podsekcja dokumentu, a nie konkretnego zapisu, natomiast właściwe powiązanie zapisu i rozrachunku jest realizowane poprzez podanie w sekcji zapisu oraz w sekcji *Transakcja* lub *Rozliczenie* tej samej wartości pola *\$idDlaRozliczen*. Program wprowadzając do baz danych programu informacje o rozrachunkach pobierze wartość pola *\$idDlaRozliczen* w sekcji rozrachunków i odszuka tę sekcję zapisów, która będzie miała tę samą wartość w polu *\$idDlaRozliczen*. Wartość ta musi być większa od 0. Jeśli zapis z takim numerem nie zostanie odnaleziony, wówczas takie rozrachunki zostaną zignorowane (ze stosownym komunikatem w oknie informacyjnym)

Przykład sekcji wyjściowej

```

Dokument{
    ...
    Zapis{
        $Kwota=12,89
        $Konto=201-1
        $idDlaRozliczen = 2875
    }
    ...
}

```

```

    }
    ...
    Transakcja{
        $Termin=14
        $idDlaRozliczen = 2875
        ...
    }
}

```

Wartości pola *\$idDlaRozliczen* w zapisach muszą być większe od zera oraz unikalne w ramach jednego dokumentu. Wartości tego pola dla poszczególnych zapisów nie muszą być kolejne ani nie muszą się rozpoczynać się od 1 (czyli mogą od wyższej wartości).

Rozróżnia się dwa zasadnicze przypadki powiązania rozrachunków z zapisami lub dokumentami. Zależą one od definicji rodzaju dokumentu, uzgodnionego w programie FK dla określonego typu dokumentu wczytywanego.

Dokumenty typu DP, WB, RK, DIM, DEX, FKZ, FKS, RUZ, RUS

Dla tych dokumentów każdy rozrachunek jest związany z konkretnym zapisem w dokumencie. Powiązaniem pomiędzy zapisem a transakcją lub rozliczeniem jest albo umieszczenie rozrachunku jako podsekcji zapisu, albo zgodna wartość pola *\$idDlaRozliczen* w zapisie i odpowiadającej mu sekcji rozrachunków umieszczonych jako podsekcje dokumentu.

Dokumenty typu FVZ, FVS, RUZ, RUS

Dla tego typu dokumentów zasada wiązania rozrachunków z dokumentem i zapisami jest zależna od strony dla której wypełniane są rozrachunki.

Dla strony kontrahenta (Wn dla FVS, RUS, Ma dla FVZ, RUZ)

Dla faktur i rachunków uproszczonych wszystkie zapisy występujące po stronie kontrahenta (czyli strona kwoty brutto: WN dla sprzedaży, MA dla zakupu) i wchodzące na to samo konto są sumowane i do rozrachunków wchodzi jako jedna kwota zbiorcza. Kwota ta nie jest związana z żadnym konkretnym zapisem, a z całym dokumentem (jedną jego stroną). W takim przypadku sekcja rozrachunków musi być umieszczona jako podsekcja dokumentu, a pole *\$idDlaRozliczen* w tych sekcjach przyjmuje wartość -1, co oznacza rozrachunki zbiorcze.

Przykład sekcji wyjściowej

```

Dokument{
    ...
    Zapis{
        $Kwota=12,89
        $Konto=201-1
    }
    ...
    Zapis{
        $Kwota=245,12
        $Konto=201-1
    }
    ...
    Transakcja{
        $Termin=14
        $idDlaRozliczen = -1
        ...
    }
}

```

Dla strony przeciwnej niż strona kontrahenta (Ma dla FVS, RUS, Wn dla FVZ, RUZ)

Dla tej strony zasady łączenia rozrachunków z zapisami są takie same, jak dla wszystkich pozostałych dokumentów, czyli sekcje rozrachunków umieszczamy jako podsekcje zapisu, albo zapewniamy zgodną wartość pola \$idDlaRozliczen w zapisie i odpowiadającej mu sekcji rozrachunków umieszczonych jako podsekcje dokumentu.

1.7 Warianty sekcji wyjściowej Dokument wraz z jej podsekcjami.

Dokument DP oraz WB

Dla dokumentu typu DP oraz WB w sekcji dokumentu muszą zostać utworzone co najmniej dwie podsekcje zapisów: dla strony Wn oraz Ma. Maksymalna liczba podsekcji Zapis zależy oczywiście od liczby zapisów księgowych w importowanym dokumencie.

1.8 Sposób budowania XSLT dla szablonów dla profili EDS.

Pobieranie dokumentów z aplikacji EDS polega na imporcie danych z pliku xml i na wstępie stransformowaniu.

Transformację XMLi za pomocą XSLT jest przetwarzaniem równoległym a sam język XSL nie jest językiem strukturalnym. Podstawowym tagiem XSL jest **xsl:template**, dla którego w parametrze **match** określa się tagi wejściowego XMLa, które będą podlegały przetworzeniu.

Przykładowo jeżeli dokument wejściowy XML zawiera tag <ATRYBUT> a w XSLT jest określony <xsl:template match="ATRYBUT"> to transformacja zostanie wywołana dla każdego taga <ATRYBUT> niezależnie od jego pozycji w hierarchicznej strukturze wejściowego XML.

Tag **xsl:template** w XSLT oprócz odpowiedniego sformatowania nowego pola zawierają przeważnie tag **xsl:value-of**, dla którego w parametrze **select** występuje zapytanie XPath w kontekście danego template'a.

Inne przydatne tagi to **xsl:output** określający formatowanie wyjścia, oraz **xsl:choose** oferujący możliwość przetwarzania warunkowego, **xsl:variable** umożliwiający definiowanie zmiennych.

1.9 Wykorzystanie XPath w XSLT.

XPath oferuje skróconą liczbę wbudowanych funkcji i operatorów oraz umożliwia wykorzystanie funkcji zewnętrznych. XPath nie oferuje statycznej kontroli typów. XPath nie oferuje współpracy z XSD i nie zapewnia sprawdzania poprawności typów ani nazw tagów/atrybutów przetwarzanych plików XML. XPath oferuje konwencje informowania o błędnym przetworzeniu, często wynikającym ze zmiany specyfikacji pliku wejściowego, błędów programistycznych lub literówek, w pliku wyjściowym. Na przykład dodając daną będącą liczbą do stringa otrzyma się w tagu wyjściowym wynik NaN.

Wykorzystane operatory i funkcje numeryczne:

- + (dodawanie)
- div (dzielenie)
- sum() - sumowanie zbioru nodów

Wykorzystane funkcje na ciągu znaków:

- substring() – obcięcie stringa
- translate() – mapowanie pojedynczych znaków ze stringa wejściowego na pojedyncze znaki wyjściowe

Wykorzystane operatory i funkcje na nodach:

- / - element główny dokumentu
- ../ - element nadrzędny względem kontekstu
- [] – element spełniający warunek
- position() – funkcja zwracająca pozycję względem node'a nadrzędnego
- text() – funkcja zwraca wartość tekstową wewnątrz node'a.

Wykorzystane operatory logiczne:

- = (prawda jeżeli równe)
- != (prawda jeżeli różne)

2 Uzgodnienia

Uzgodnienia są to zapamiętywane w programie połączenia pewnych informacji pochodzących od nadawcy danych oraz informacji zawartych w programie Finanse i Księgowość. Uzgodnienia są dokonywane zasadniczo przez użytkownika importu specjalnego, choć w pewnych przypadkach mogą być dokonane automatycznie przez program. Przykładem uzgodnienia może być uzgodnienie pozycji kartotek. Każdy nowy kontrahent wczytywany podczas importu musi być skojarzony z odpowiednim kontrahentem w kartotece programu FK. Może wystąpić tu kilka przypadków, np. kontrahenta jeszcze nie ma w kartotece FK, istnieje i ma wprowadzone identyczne dane, istnieje ale ma inne dane, np. inaczej zapisana nazwa kontrahenta, inaczej zapisany adres (np. Łódź –Lodz itp.) W przypadkach niejednoznacznych program otwiera kartotekę kontrahentów, pokazuje okno z danymi importowanymi i użytkownik musi zdecydować, której pozycji w programie FK odpowiada importowany kontrahent. Dokonane w ten sposób połączenie danych importowanych i danych w programie FK nazywane jest uzgodnieniem i jest przez program zapamiętywane. Przy każdym następnym wystąpieniu w danych importowanych tego uzgodnionego kontrahenta powoduje, że program korzysta z dokonanego wcześniej uzgodnienia i bez pytania użytkownika używa właściwej pozycji kartoteki z programu FK.

Wszystkie dokonane uzgodnienia mogą być zweryfikowane w oknie Uzgodnienia. Można sprawdzić poprawność dokonanych uzgodnień i ewentualnie usunąć te uzgodnienia, które zostały dokonane nieprawidłowo (lub jeśli musimy zmienić np. przypisania do typów dokumentów w programie FK). Usunięcie uzgodnień nie ma wpływu na dokumenty (kartoteki) już zaimportowane i znajdujące się w programie FK, dopiero podczas kolejnego importu danych program poprosi o ponowne dokonanie usuniętych uzgodnień.

Uzgodnienia dla poszczególnych profili

Uzgodnienia danych są związane z konkretnym profilem nadawcy, który jest używany do wczytywania danych. Oznacza to, że uzgodnienia dokonane dla jednego nadawcy (profilu) nie są widoczne podczas wczytywania danych innego nadawcy. Jest to oczywiste, gdyż różni nadawcy mogą mieć w swojej kartotece pod tym samym identyfikatorem zupełnie różnych kontrahentów. Dla każdego nadawcy uzgodnienia należy zrobić oddzielnie.

UWAGA: z tego samego powodu nie wolno używać jednego profilu do wczytywania danych od kilku nadawców, jeśli mają oddzielne kartoteki kontrahentów (np. oddziały które nie pracują na wspólnej sieciowej firmie).

Dla każdego nadawcy (posiadającego oddzielną kartotekę kontrahentów) musimy w programie FK zdefiniować oddzielny profil.

Dokonane uzgodnienia są przechowywane w tabeli bazy danych osobno dla każdego profilu i użytkownika.

Uzgodnienia dla kilku szablonów w jednym profilu

Zdarza się często, że dane importowane są umieszczane w kilku plikach, na przykład oddzielnie dane kontrahentów, a oddzielnie dane dokumentów (przykładem jest format FK2.0). Tych oddzielnych plików nie można wczytać przy pomocy oddzielnych profili importu, gdyż z powodów opisanych wcześniej (oddzielnych plików uzgodnień dla poszczególnych profili) uzgodnienia dokonane podczas importu kontrahentów nie byłyby widoczne i nie mogłyby zostać użyte podczas importu dokumentów, a więc niemożliwe byłoby określenie kontrahenta i jego numeru konta na dokumencie. Dane z kilku plików muszą być wczytane przy użyciu jednego profilu. Aby tego dokonać, należy użyć polecenia `#execute` lub `Execute()`, które umożliwiają wywołanie z jednego szablonu procesu wczytywania danych z innego pliku przy pomocy innego szablonu. W takiej sytuacji wszystkie uzgodnienia dokonane w jednym szablonie są widoczne we wszystkich innych szablonach (zarówno w „głównym” jak i tych wykonywanych poleceniem `Execute()`), oczywiście cały czas w ramach jednego profilu nadawcy.

W szczególnym przypadku można ten mechanizm zastosować do kilkakrotnego wczytania jednego pliku danych kolejno za pomocą kilku szablonów. Może to być potrzebne gdy np. w jednym pliku mamy zarówno kontrahentów i dokumenty, a najpierw musimy dokonać uzgodnień wszystkich kontrahentów, a dopiero później zaimportować dokumenty.

2.1 Uzgodnienia Importu z EDS

Uzgodnienia kontrahentów z EDS następuje automatycznie na podstawie wysyłanego przez EDS pola GUIDEDS. Jeżeli w programie FK znajdzie się kontrahent z taką samą wartością pola GUID to automatycznie dla dokumentu importowanego przypisywany jest znaleziony kontrahent. W przypadku braku kontrahenta w programie FK następuje próba poszukiwania po polu NIP. Jeżeli w programie FK znajdzie się kontrahent z taką samą wartością pola NIP to automatycznie dla dokumentu importowanego przypisywany

jest znaleziony kontrahent. W przypadku nie znalezienia kontrahenta w programie FK po polach GUID lub NIP użytkownik ma możliwość uzgodnienia ręcznego kontrahenta, przez wskazanie uzgodnienia na otwartej liście kontrahentów lub dodanie nowego kontrahenta. UWAGA: Dodany nowego kontrahenta spowoduje, że w programie FK zapisana będzie wartość z pola GUID. Każdy następny dokument z kontrahentem o tej samej wartości w polu GUID wymusi automatyczne przypisanie kontrahenta poprzednio dodanego. W przypadku uzgodnienia ręcznego kontrahenta istniejącego w FK nie jest przypisywany GUID.

3 Informacje zwrotne

Po zakończeniu importu specjalnego mogą zostać odesłane do nadawcy pewne informacje związane z właśnie wczytanymi danymi. Mogą to być dwie zasadnicze kategorie informacji:

- informacje o uzgodnieniach dokonanych podczas importu danych
- informacje o nie rozliczonych transakcjach, związanych z kontrahentami którzy wystąpili we wczytywanych danych od tego konkretnego nadawcy.

Informacje o uzgodnieniach

Import specjalny może stworzyć plik danych zwrotnych zawierający dowolną kombinację poniższych informacji:

- uzgodnienia kartoteki kontrahentów
- uzgodnienia kartoteki pracowników
- uzgodnienia kartoteki urzędów
- uzgodnienia kartoteki rachunków bankowych
- uzgodnienia kartoteki kontrahentów incydentalnych
- uzgodnienia typów dokumentów
- uzgodnienia definicji rejestrów VAT

Odesłaniu podlegają informacje o połączeniach dokonanych od ostatniego wysłania informacji zwrotnych. Istnieje też możliwość ponownego wysłania dowolnych poprzednich informacji zwrotnych,

Informacje transakcjach

Możliwość przesyłania do nadawcy informacji o transakcjach z kontrahentami, którzy zostali zaimportowani od danego nadawcy została wprowadzona w wersji FK 4.0c.

Możliwy jest wybór trzech poziomów szczegółowości danych:

Tylko informacje o sumie nierozliczonych należności i zobowiązań z danym kontrahentem.

Dodatkowo oprócz informacji o sumie nierozliczonych należności i zobowiązań z danym kontrahentem zwracana jest lista wszystkich transakcji z tym kontrahentem. Można tu określić dodatkowo, czy mają być umieszczone wszystkie transakcje, czy tylko transakcje nierozliczone.

Dodatkowo oprócz informacji o poszczególnych transakcjach zwracane są wszystkie częściowe rozliczenia transakcji.

W wariantcie najprostszym w pliku uzgodnień wysyłane są następujące informacje:

Rodzaj kartoteki

Wartość pola wybranego jako unikalny identyfikator kartoteki u nadawcy (klucz)

Suma należności

Suma zobowiązań

Przykładowa sekcja w pliku uzgodnień wygląda następująco:

```
Rozrachunki{  
  Kartoteka =K  
  klucz =103  
  Wn =340.00  
  Ma =7890.00  
}
```

W wariantcie z przesyłaniem informacji o wszystkich nierozliczonych transakcjach, dla każdej nierozliczonej transakcji z danym kontrahentem wysyłane są następujące dane:

Strona transakcji

Kwota początkowa transakcji w zł oraz ewentualnie waluta i kwota w walucie

Numer własny dokumentu z którego pochodzi transakcja

Termin płatności

Kwota pozostała do rozliczenia w zł oraz walucie
Znacznik, czy transakcja jest zaliczką

Przykładowa sekcja danych wygląda następująco:

```
Rozrachunki{
  Kartoteka =K
  klucz =103
  Wn =340.00
  Ma =7890.00
  transakcja{
    strona ="Wn"
    kwota =937.51
    kwotaWal =0.00
    waluta =" "
    dokument ="WB 39/01/11"
    termin =1998-03-13
    rozliczona =1
    doRozlicz =34.00
    doRozliczWal =0.00
    zaliczka =0
  }
  ....
}
```

Dla wariantu najbardziej rozbudowanego dodatkowo dla każdej transakcji jest dodawana informacja o wszystkich częściowych rozliczeniach tej transakcji.

Informacje o każdym rozliczeniu są następujące:

Strona rozliczenia

Kwota rozliczenia w zł oraz w walucie

Numer własny dokumentu rozliczającego

Data dokumentu rozliczającego

Przykładowa sekcja zawierająca maksymalnie szeroki zakres informacji wygląda następująco:

```
Rozrachunki{
  Kartoteka =K
  klucz =103
  Wn =340.00
  Ma =7890.00
  transakcja{
    strona ="Wn"
    kwota =937.51
    kwotaWal =0.00
    waluta =" "
    dokument ="WB 39/01/11"
    termin =1998-03-13
    rozliczona =1
    doRozlicz =34.00
    doRozliczWal =0.00
    zaliczka =0
  }
  rozliczenie{
    strona ="Ma"
```

```

    kwota =903.51
    kwotaWal =0
    dokument ="WB 39/01/11"
    data =1998-03-19
}
....
}
.....
}

```

Do określenia formatu pliku z informacjami zwrotnymi oraz ustalenia zakresu informacji zwrotnych służą polecenia szablonu: #RETURN lub RETURN(), opisane w dalszej części opracowania.

3.1 Rozszerzenie od wersji 2018.2

Wraz z wprowadzeniem wersji 2018.2 ERP możliwe jest także pozyskanie informacji o dokumentach zapisanych do bufora w trakcie importu i powiązania ich z własnym systemem identyfikacji. Informacja taka bywa używana w rozwiązaniach „trzecich” (integratorów) kiedy to po zaimportowaniu dokumentów informacja o dalszych zmianach dokumentu już w systemie FK jest wykorzystywana przez system wysyłający dokumenty. Informację o identyfikatorze [id] zapisanego dokumentu można odczytać bezpośrednio z okna komunikatów postępu importu specjalnego poprzez np. wykonanie raportu z okna importu. Potrzebne jest w tym celu spełnienie następujących warunków:

1. Nadawca danych powinien zapewnić umieszczenie w danych własnego identyfikatora dokumentu – używanego w swoim systemie – może być to dowolny ciąg znaków (typu guid, id, etc.) unikalny w ramach pojedynczej sesji importu.
2. Wartość ta powinna być poprzez instrukcję w szablonie podstawiana do pola \$dokident, np.
\$dokIdent = [dokIdent]
3. Aby uzyskać w oknie importu informację o powiązaniu należy w szablonie użyć definicji #INFO_DOKID "1". Przebieg z wartością "1" spowoduje wyświetlanie powiązania, przy wartości "0" program tych powiązań nie wyświetli. Komunikat ma postać:
[Dokument o kluczu nadawcy:0123456 - został zapisany pod id:18372](#)
Wiersze w oknie postępu importu przechowujące te powiązania mają wydzielony numer typu danych równy 7 (zmienna TypDanych).

Przykład użycia można obejrzeć w szablonie hmffk.ams.

4 Format danych wejściowych

4.1 Definiowanie formatu plików

4.1.1 Informacje ogólne

Import specjalny obsługuje dane w różnorodnych formatach. Razem z programem dostarczanych jest wiele definicji popularnych formatów danych, które mogą być używane natychmiast. Dodatkowo użytkownik ma możliwość zdefiniowania (opisania) innego, własnego formatu danych. Dzięki temu używając standardowego programu FK można zdefiniować taki szablon, który wczyta dane z innego programu bez konieczności dokonywania kosztownych i czasochłonnych zmian w obydwu programach. Istnieje oczywiście kilka założeń, które muszą być spełnione, aby dane dały się opisać własną definicją formatu:

Plik danych musi być w formacie tekstowym (plik ASCII)

Oddzielne pliki nie mogą zawierać danych powiązanych relacyjnie za sobą, np. jeden plik zawiera nagłówki dokumentów, drugi zapisy księgowe, a trzeci rejestry VAT lub rozrachunki.

Wyjątkiem od tej reguły są kartoteki, które mogą być umieszczane w innych plikach niż dokumenty księgowe. Format jednego pliku powinien być jednorodny (tzn. nie może być jednych danych w formacie CDF, a innych w formacie SDF)

Musi istnieć jednoznaczny sposób opisu podstawowych elementów formatu, czyli:

- początku sekcji
- końca sekcji
- nazwy pola (o ile występuje)
- wartości pola

- komentarzy (o ile występują)

Do opisu elementów formatu wykorzystano tzw. wyrażenia regularne, czyli pewien sposób opisu właściwości analizowanego tekstu, tak aby opisać jego cechy a nie konkretną wartość. Najprostszymi wyrażeniami regularnymi były znaki typu *(dowolny ciąg znaków) oraz ? (dowolny jeden znak) używane w systemie DOS do określenia grupy plików.

Dzięki rozbudowanemu mechanizmowi wyszukiwania niejednoznacznego można tworzyć nawet bardzo skomplikowane opisy, w stylu:

Pole danych składa się z tekstu umieszczonego w parze nawiasów trójkątnych, przy czym nazwą pola jest tekst do znaku dwukropka, a wartością jest tekst po znaku dwukropka, przy czym dodatkowo mają być ignorowane (nie są danymi) wszystkie białe znaki pomiędzy znakiem dwukropka a zamykającym nawiasem.

Co odpowiada przykładowym danym:

```
<kwota: 143765,84 >
<czas:12:28.56>
```

i opisywane jest wyrażeniem

```
<{*}/: :b{*}:b>
```

Pełny opis wyrażeń regularnych można znaleźć w pliku pomocy do języka raportów (AMBASFK.CHM), który dostarczany jest standardowo wraz z programem Finanse i Księgowość.

Pomimo niewielu elementów składających się na definicję formatu, opisywanie własnych formatów danych wymaga sporej wiedzy i często pomysłowości. Można w firmie Sage sp. z o.o. zamówić płatną usługę tego rodzaju.

4.1.2 Elementy opisu formatu

Sekcja

Sekcja jest elementarną porcją informacji, wczytywanej i przetwarzanej jako całość. W zależności od formatu może to być jedna linia w pliku (format CDF lub SDF), albo też kilka kolejnych linii (np. format AMS). Do prawidłowego podzielenia pliku wejściowego na sekcje (czyli np. kolejne dokumenty) należy opisać w jaki sposób rozpoznawany będzie początek sekcji, oraz w jaki sposób rozpoznawany będzie koniec sekcji.

W najprostszym przypadku, dla formatu SDF lub CDF:

początkiem sekcji jest początek linii, opisuje to wyrażenie ^

końcem sekcji koniec linii, opisuje to wyrażenie \$

Dla bardziej skomplikowanego przypadku AMS

początkiem sekcji jest dowolna nazwa po której występuje otwierający nawias klamrowy, opisuje to wyrażenie ^[\t]#[*][\t]#/{*\$-

końcem sekcji jest zamykający nawias klamrowy, opisuje to wyrażenie ^*/}\$-

Każdy opis własnego formatu musi mieć zdefiniowany opis początku i końca sekcji.

Przy definiowaniu początku sekcji należy zwrócić uwagę na dwa elementy, mianowicie nazwę sekcji, którą będzie wyrażenie znajdujące się wewnątrz nawiasów klamrowych w wyrażeniu opisującym początek sekcji. Oczywiście wyrażenie to występuje tylko dla formatów zawierających jawnie nazwę sekcji.

Przykładowo jeśli plik zawiera pola oddzielone średnikami, a pierwsze pole zawiera symbol wyróżniający typ danych, np. typ dokumentu:

```
FAKTURA; 0; 123,56; 22; „Sprzedaż towarów”
```

```
KASA; 0; 5857,54; 201-1-1285; „Zapłata za fakturę”
```

W takim wypadku możemy stworzyć sekcje wejściowe o różnych nazwach, opisując początek sekcji następująco:

```
^{*};-
```

co oznacza, że nazwą sekcji będzie wartość pierwszego pola, czyli dowolny ciąg znaków od początku linii do pierwszego średnika.

Drugim ważnym elementem wyrażenia regularnego jest znak – (minus), który określa, od którego miejsca w analizowanej linii wejściowej ma być kontynuowane poszukiwanie następnych elementów pliku (podsekcji oraz pól)

W powyższym przykładzie następną analizowaną informacją w linii będzie ciąg znaków po pierwszym średniku (minus jest po średniku). Znak minus może być umieszczony w dowolnym miejscu wyrażenia, stosownie do potrzeb. Jeśli np. chcielibyśmy pominąć drugie kolejne pole (w przykładzie ma wartość 0), to można byłoby zapisać

```
^{*};*;-
```

a jeśli nazwą miałyby być tylko dwie pierwsze litery z pierwszego pola, a reszta tego pola już daną (może nie ma to wielkiego sensu, ale dobrze ilustruje działanie minusa!), zapisalibyśmy:

```
^{??}-*;
```

W specyfikacji końca sekcji nie ma już potrzeby oznaczania nazwy sekcji i całe wyrażenie służy tylko identyfikacji końca sekcji. Definicja końca sekcji w pliku danych nie musi być związana z końcem linii, nawet jeśli rzeczywiście jedna linia zawiera jedną sekcję. Przykładem może być format AM, w którym znakiem końca sekcji jest znalezienie początku nowej sekcji lub słowa **koniec**.

Tak może wyglądać przykładowy plik danych

```
„DOK”; 1; 1998-12-03
„VAT”; 2746,56; 0; 22; A
„DOK”; 2; 1998-12-04
„VAT”; 2746,56; 0; 22; A
```

Koniec.

Linia zawierająca na początku „DOK” jest sekcją dokumentu, a kolejna linia „VAT” jest jej podsekcją.

Definicja końca sekcji dokumentu wygląda następująco

```
(^-"{DOK}") | (^#Koniec.-)
```

mamy tu aż dwa minusy!

Jeśli w pliku danych zostanie odnaleziony tekst „DOK”, to oczywiście zakończy to poprzednią sekcję dokumentu, ale znak minus jest przed „DOK”, więc ten tekst nie zostanie „zjedzony” jako znak końca sekcji, tylko pozostanie do dalszej analizy i zostanie rozpoznany jako początek nowej sekcji. Trochę inaczej będzie jeśli zostanie rozpoznane słowo Koniec. W tym przypadku minus jest na końcu tego słowa, więc zostanie ono pominięte i nie będzie już dalej analizowane.

Wartości

Dane zawarte w pliku danych zostały umownie podzielone na dwie kategorie: „dane liczbowe” oraz „dane tekstowe”. W zasadzie rozsądniejszym podziałem mogło być „dane w cudzysłowie” oraz dane „dane bez cudzysłowu”, bo de facto do tego sprowadza się ten podział.

Pod nazwą dane liczbowe ukrywają się wszystkie dane, które nie są umieszczane w cudzysłowie, czyli:

- liczby
- daty
- kwoty
- numery nip
- kody pocztowe

teksty, które nie zostały podane w cudzysłowie

Natomiast „dane tekstowe” oznaczają teksty umieszczone wewnątrz znaków cudzysłowu. Rozróżnienie typów danych ma na celu tylko i wyłącznie usunięcie znaków cudzysłowu podczas wczytywania danych tak, aby pole wejściowe dla szablonu zawierało tylko „czyste” dane.

Należy pamiętać, że dla szablonu wszystkie dane z pliku wejściowego są tekstami, niezależnie czy są umieszczone w cudzysłowie, czy też nie. Ponieważ opis wartości numerycznej i tekstowej jest dokonywany poprzez wyrażenia regularne, mówiąc „cudzysłów” mam na myśli dowolne znaki znajdujące się na początku i końcu tekstu. Mogą to być znaki cudzysłowu „”, apostrofów ’ lub dowolne inne, np. nawiasów, prefiksów itd.

Przykładowe pola w formacie SDF:

```
"Zakup towarów";1999-12-28;7376,46; "Holding S.A."
```

Definicja wartości numerycznej

```
^|;{*}-;|§
```

którą należałoby czytać:

wartość numeryczna zaczyna się od początku linii lub od znaku średnika, po którym następuje właściwa wartość będąca dowolnym ciągiem znaków o długości 0 lub większej, zakończona średnikiem lub znakiem końca linii.

Definicja wartości tekstowej

```
^|;"{*}"-;|§
```

którą należałoby czytać:

wartość tekstowa zaczyna się od początku linii lub od znaku średnika, po którym następuje znak cudzysłowu (pomijanego) oraz właściwa wartość będąca dowolnym ciągiem znaków o długości 0 lub większej, zakończona znakiem cudzysłowu (pomijanym) oraz średnikiem lub znakiem końca linii.

Jak widać w obydwu przykładach znak minus znajduje się przed średnikiem kończącym pole, tak więc analiza następnego pola rozpocznie się od znaku średnika, co spełni warunek początkowy dla następnej

definicji pola (pierwsze pole zaczyna się od początku linii, kolejne pole zaczyna się od średnika).

Bardziej złożonym przykładem jest wartość z nazwą pola

Nazwa kontrahenta = „Holding S.A.”

Oraz opis tego pola:

```
/=:b{*}" :b $-
```

który należy czytać:

wartość tekstowa zaczyna się po znaku równości, po którym może znajdować się dowolna ilość białych znaków. Zaczyna się znakiem cudzysłowu (pomijanym). Jest ona dowolnym ciągiem znaków o długości 0 lub większej, zakończona znakiem cudzysłowu (pomijanym) po którym mogą znajdować się do końca linii tylko białe znaki (spacje, tabulatory etc.)

Nazwy pól

Dla formatów z nazwą pola można określić nazwę pola, którą później będzie można używać w szablonie do identyfikacji danych. Nie jest to wymóg konieczny i można do pól odwoływać się poprzez ich numer kolejny, ale zwykle w formatach z nazwami pól kolejność pól nie jest sztywno ustalona i może ulec zmianie. Jest to jedna z zalet formatu z nazwami pól. Nazwy pól opisuje się analogicznie jak wartości pól, ale w tym przypadku nie ma rozróżnienia pomiędzy polami numerycznymi i tekstowymi.

Przykład

```
Nazwa kontrahenta = „Holding S.A.”
```

```
Oraz opis nazwy pola
```

```
^:b{*}:b-/=
```

Co oznacza:

Nazwa pola zaczyna się od początku linii, z pominięciem początkowych białych znaków. Jest to ciąg dowolnych znaków o długości 0 lub więcej znaków, po których występuje dowolna ilość białych znaków (pomijanych) i znak równości.

Warto zwrócić uwagę na położenie znaku minus, który znajduje się przed znakiem równości i powoduje to, że analiza dalszej części linii będzie kontynuowana od znaku równości razem z tym znakiem – porównaj z definicją wartości tekstowej dla pól z nazwami: zaczyna się od znaku równości.

Komentarze

W pliku danych mogą znajdować się linie, które nie zawierają istotnych informacji dla importu danych, a tylko uwagi i komentarze przeznaczone dla osób które będą przeglądać plik danych. Zwykle uwagi takie nie spełniają wymagań co do formatu właściwych danych i naruszają w pewien sposób strukturę pliku. W każdym razie nie można ich potraktować i wczytać jako dane, a odrzucić w szablonie jako dane nieistotne. Muszą one zostać zidentyfikowane i odrzucone już na etapie analizy formatu pliku. Służy temu definicja (opisy) komentarza.

Przykład komentarza liniowego dla formatu SDF (znakiem komentarza jest średnik na początku linii):

```
;to jest komentarz w pliku SDF, ta linia zostanie zignorowana. Następna linia również
```

```
;RK;"Zapłata za fakturę";12387,98;22;201-1;1999-12-30
```

Oraz bardzo prosta definicja tego komentarza:

```
^;
```

średnik znajdujący się na początku linii

Przykład komentarza liniowego w formacie AMS, tworzą go dwa znaki ukośnika //

```
//komentarz w formacie AMS, kolejna linia też zawiera komentarz po polu danych  
kwota = 876,98 //bez podatku VAT
```

Oraz definicja tego komentarza:

```
////*
```

Dowolny ciąg znaków po dwóch znakach ukośnika

Proszę zwrócić uwagę, że w definicji tego komentarza znajdują się cztery ukośniki. Wynika to z faktu, że ukośnik ma specjalne znaczenie w wyrażeniach regularnych, i aby zaznaczyć, że chodzi nam o pojedynczy znak ukośnika a nie specjalne znaczenie ukośnika w wyrażeniach regularnych, musimy wpisać dwa ukośniki.

Komentarz liniowy zawsze obowiązuje od początku komentarza do końca linii.

4.1.3 Pliki polimorficzne

Często bywa tak, że jeden plik danych zawiera informacje różnego rodzaju, zapisane w trochę różny sposób. Przykładowo w pliku o formacie SDF część linii zawiera informację o dokumentach księgowych, a część o kontrahencie. Przykładem jest format AM, w którym jedna linia SDF zawiera dane dokumentu, a kolejne linie dane zapisów księgowych, rejestrów VAT i rozrachunków. W takich przypadkach należy zdefiniować kilka kolejnych opisów różnych sekcji (początku i końca sekcji, formatu danych itd.). Każdy taki komplet danych opisuje jeden rodzaj sekcji wejściowych. W każdej sekcji mogą być dane w innym formacie, przykładem jest format POLKA, który w jednym rodzaju sekcji ma nazwy pól oddzielone dwukropkiem, a w innej oddzielone znakiem równości.

4.1.4 Kolejność analizowania elementów formatu

Do pełnego zrozumienia zasad definiowania własnych formatów danych niezbędna jest znajomość reguł rządzących analizą pliku wejściowego. W poprzednich punktach opisane zostały zasady rozpoznawania poszczególnych elementów pliku wejściowego, teraz omówię kolejność w jakiej poszczególne elementy są analizowane.

Plik danych jest analizowany linia po linii. Jako linię rozumie się ciąg znaków ASCII do znaku końca linii, czyli CR LF (kody 10 oraz 13).

pierwszą sprawdzaną rzeczą jest komentarz. Jeśli zostanie odnaleziony wówczas cała linia do końca jest ignorowana i wczytywana jest następna linia z pliku danych.

Sprawdzany jest początek sekcji. Jeśli definicja formatu zawiera więcej niż jedną definicję sekcji, wówczas sprawdzane są kolejno wszystkie definicje aż do napotkania pierwszej zgodnej definicji.

Jeśli natrafiono na początek sekcji, to tworzona jest sekcja wejściowa.

Po zidentyfikowaniu początku sekcji następnym sprawdzanym elementem jest koniec sekcji.

Jeśli nie ma końca sekcji, to sprawdzany jest początek następnej sekcji (podsekcji). Jeśli znaleziono początek sekcji to tworzona jest podsekcja sekcji wejściowej i analiza powraca do punktu 2.1.1

Jeśli natrafiono na koniec sekcji (komplementarny do ostatnio znalezionej sekcji) to zamykana jest podsekcja wejściowa i przechodzimy do punktu 3

wczytywane jest pole danych, używając definicji pól zgodnej z ostatnio odnalezioną sekcją. Dane są umieszczane w ostatniej otwartej sekcji (podsekcji) wejściowej

wracamy do punktu 1.

Powyższa kolejność jest zachowana dla wszystkich formatów predefiniowanych, oraz dla formatów własnych użytkownika.

4.1.5 Definiowanie formatu w pliku IMPEXP.INI

W pliku IMPEXP.INI, znajdującym się w katalogu instalacyjnym programu Finanse i Księgowość, można zdefiniować własne, dodatkowe formaty pliku wejściowego. Struktura tego pliku jest analogiczna jak innych typowych plików INI w systemie Windows (np. pliku WIN.INI), czyli zawiera nazwane sekcje, zawierające nazwane wejścia z określoną wartością. Pierwszą istotną sekcją w pliku IMPEXP.INI jest sekcja `_RD_FORMAT`, mająca następującą postać:

```
[_RD_FORMAT]
SDF=1
AM=2
POLKA=2
AMS=1
INI=1
COMMA=1
SUBIEKT=3
```

Każde wejście w tej sekcji składa się z nazwy formatu, oraz po znaku równości liczby właściwych sekcji opisujących format pliku. Aby dodać własny format, należy dodać w tej sekcji nazwę kolejnego formatu i liczbę jego podsekcji definiujących. Właściwy opis Formatu znajduje się w sekcjach opisu. Zasady tworzenia

tych sekcji są następujące: nazwę sekcji definiującej budujemy dodając przed nazwą formatu człon `_RD_FORMAT`, a po nazwie formatu znak podkreślenia oraz numer kolejny sekcji w ramach tego formatu. W poniższym przykładzie podano sekcje definiujące dla formatu SDF

```
[_RD_FORMAT_SDF_0]
; komentarze, zmienne globalne i ustawienia translatora
KOM_LIN_POCZ=(^;) | (^Koniec.)

[_RD_FORMAT_SDF_1]
; opis sekcji musi zawierać początek i koniec sekcji oraz definicje pól
SEKCJA_POCZATEK=^-{*}[~ \t]
SEKCJA_KONIEC=$
WARTOSC_LICZBOWA=^|;{*}-;|$
WARTOSC_TEKSTOWA=^|;"{*}"[ \t]#-;|$
```

Sekcja o numerze 0 jest zarezerwowana dla opisu komentarzy, zmiennych globalnych oraz ustawień translatora.

Kolejne sekcje, o numerach od 1 aż do numeru podanego w sekcji definicji nazwy formatu zawierają opis kolejnych formatów sekcji, w kolejności ich analizowania podczas wczytywania pliku wejściowego

Przykładowo chcemy utworzyć dodatkowy format pliku o następującej postaci:

```
<sekcja>
  <pole=wartość>
  <pole2=wartość2>
</sekcja>
<REM> wszystko po słowie rem w nawiasach jest komentarzem
```

Dopisujemy nową nazwę formatu o nazwie NOWY

```
[_RD_FORMAT]
SDF=1
AM=2
POLKA=2
AMS=1
INI=1
COMMA=1
SUBIEKT=3
NOWY=1
```

Oraz tworzymy sekcje definiujące:

```
[_RD_FORMAT_NOWY_0]
; komentarze, zmienne globalne i ustawienia translatora
KOM_LIN_POCZ=<REM>

[_RD_FORMAT_NOWY_1]
; opis sekcji musi zawierać początek i koniec sekcji oraz definicje pól
SEKCJA_POCZATEK=<{[~=]++}>-
SEKCJA_KONIEC=<//[~=]++>-
WARTOSC_LICZBOWA=/={*}>-
WARTOSC_TEKSTOWA=/={*}>-
POLE_NAZWA=<{*}-/=
```

Znaczenia poszczególnych nazw wejść w sekcjach definiujących są następujące:

KOM_LIN_POCZ - Opisuje sekwencję znaków, które oznaczają początek komentarza liniowego w pliku danych.

SEKCJA_POCZATEK - Opisuje sekwencję znaków oznaczającą początek sekcji w pliku danych. Wyrażenie ujęte w nawias klamrowy wydziela z całego opisu nazwę sekcji.

SEKCJA_KONIEC - Opisuje sekwencję znaków, która oznaczać będzie koniec sekcji wejściowej.

POLE_NAZWA - Opisuje nazwę pola danych w pliku wejściowym. Wyrażenie ujęte w nawiasy klamrowe wydziela samą nazwę pola z sekwencji innych znaków dodatkowych (znaku początku linii, separatorów itd.)

WARTOSC_LICZBOWA - Opisuje wartość liczbową pola. Wyrażenie ujęte w nawiasy klamrowe wydziela samą wartość pola z sekwencji innych znaków dodatkowych (znaku końca linii, separatorów itd.)

WARTOSC TEKSTOWA - Opisuje wartość tekstową pola. Wyrażenie ujęte w nawiasy klamrowe wydziela samą wartość pola z sekwencji innych znaków dodatkowych (cudzysłówów, znaku końca linii, separatorów itd.)

4.1.6 Definiowanie formatu w szablonie

Definiowanie formatu w szablonie odbywa się w podobny sposób, jak definiowanie formatu w pliku IMPEX.INI, z następującymi różnicami:

Definiując format w szablonie nie określamy nazwy formatu

W szablonie nie tworzymy sekcji, ale wszystkie opisy umieszczane jeden pod drugim

Definiowanie poszczególnych elementów sekcji musi być w określonym porządku:

- definicja początku sekcji
- definicja nazwy pola
- definicja wartości pola
- definicja końca sekcji

Dla przykładowego pliku opisanego w poprzednim punkcie definicja w szablonie wyglądałaby następująco

```
#KOM_LIN_POCZ "<REM> "  
#SEKCJA_POCZATEK "<{[~=]++}>-"  
#SEKCJA_KONIEC "<//[~=]++>- "  
#WARTOSC_LICZBOWA "/={*}>-"  
#WARTOSC TEKSTOWA "/={*}>-"  
#POLE_NAZWA "<{*}-/= "
```

4.1.7 Kiedy definiować własne formaty danych

Zasada jest bardzo prosta, tylko wtedy gdy posiadany plik z danymi nie może być zinterpretowany przy pomocy istniejących formatów. Dużo bezpieczniej jest dopasować się do jednego z gotowych, standardowych formatów niż tworzyć nowy. Standardowe formaty są właściwie niezmiennie w czasie (odmiennie niż szablony) i raczej nie będzie potrzeby zarządzania zmianami w formatach. Skorzystamy w ten sposób ze sprawdzonego i przetestowanego wzorca, a także mamy pewność, że następne wersje programu nie wniosą zmian i poprawek uniemożliwiających obsługę nowego formatu (prawdopodobieństwo tego jest niewielkie, ale przy bardzo skomplikowanych konstrukcjach może się zdarzyć). Do standardowych formatów łatwiej znaleźć opis (choćby ten podręcznik), oraz osoby które będą potrafiły go obsługiwać i poprawiać ewentualne błędy.

Często jednak nie ma innego wyjścia, gdy na przykład nie można zmodyfikować programu wysyłającego dane, a format przez niego używany jest delikatnie mówiąc „egzotyczny”. Wtedy na pewno lepiej zdefiniować nowy format, niż tworzyć dodatkowe zewnętrzne translatory formatu.

Pamiętajmy jednak, że format danych nie przesądza jeszcze kategorię o postaci danych w pliku danych nadawcy! Dotyczy on bowiem formatowania pojedynczych danych. Ostateczne rozstrzygnięcie w jaki sposób interpretować jakie dane określona jest poprzez szablon, który pobierane dane z pliku przypisuje zmiennym. I tak większość szablonów profili dostarczanych przez aplikacje Symfonii korzysta z formatu AMS.

W przypadkach rozważania nowego importu z nowego programu nadawcy zalecamy użycie formatu danych AMS.

4.2 Predefiniowane formaty

4.2.1 Uwagi ogólne

Import specjalny w programie FK obsługuje wiele różnych formatów danych wejściowych, z czego część jest dostarczana razem z programem, a wiele innych można utworzyć samodzielnie. Poniższy opis

predefiniowanych (wbudowanych) formatów opisuje tylko sposób, w jaki powinny zostać skonstruowane pliki danych, aby zostały rozpoznane jako poprawne pod względem składniowym. Nie znajdziemy w tym opisie szczegółowego znaczenia każdego pola, ani wartości jakie to pole może i powinno przyjmować, aby zostać prawidłowo zaimportowane do programu. Dane zawarte w plikach każdego formatu mogą być zupełnie dowolne i dowolnie ułożone, to dopiero szablon dokonuje takiego przekształcenia danych, aby zostały one prawidłowo zaimportowane. Tak więc format opisuje tylko cechy charakterystyczne pliku, nie opisuje jego zawartości merytorycznej. W Aby dowiedzieć się jak interpretować i pobierać dane zajrzyj do opisu języka szablonów.

4.2.2 Format SDF

Informacje ogólne

Jest to jeden z najbardziej wiekowych i popularnych formatów danych. Jest używany przez programy MS Excel, MS Access i wiele innych. W formacie tym każda linia pliku zawiera jedną porcję pól danych, oddzielonych średnikami. W ogólnym przypadku formatu SDF każda linia zawiera taką samą liczbę pól, o takim samym znaczeniu. W imporcie specjalnym każdy wiersz może zawierać różną liczbę pól, a do tego mogą mieć one w każdym wierszu zupełnie różne znaczenie. Format kryjący się pod nazwą SDF jest jednym z przypadków ogólnego formatu CSV, czyli pól oddzielonych separatorami. Format SDF w imporcie specjalnym obsługuje pola oddzielone średnikami. Jeśli zaistnieje potrzeba obsłużenia formatu z polami oddzielnymi innymi znakami (np. tabulatorem lub przecinkiem), należy utworzyć własną definicję formatu, wzorując się na definicji SDF.

Pola danych

W formacie SDF pola danych oddzielone są od siebie średnikami. Pola tekstowe mogą być ujęte w cudzysłów, albo pozostać bez cudzysłowu. Jeśli nie używamy cudzysłowu, wówczas należy zadbać, aby wewnątrz pól tekstowych nie znalazł się znak średnika, który zostanie potraktowany jako separator pól. Z tego względu bezpieczniej jest stosować teksty w cudzysłowie. Jeśli jakieś pole zawiera pustą daną (zero lub pusty tekst), można skrócić zapis takiej informacji, pisząc sam średnik. Uwaga: w ten sposób nie można „skrócić” zapisu pierwszej danej w linii, gdyż tak linia zostałaby potraktowana jak komentarz.

Przykład:

```
"FV";1999-03-18; 1999-03-19;436576,56; "Holding S.A. ";233-22-33-234;01-223;"Warszawa"
```

Nazwy pól

Do wersji FK 4.0 jedynym sposobem nazywania pól w formacie SDF jest podawanie jako nazwy numeru kolejnego pola w wierszu danych. Od wersji 4.0 można używać pierwszy wiersz jako nazwy pól. Służy temu dyrektywa:

```
#NAZWY_POL „TAK”
```

Oczywiście jeśli używamy tej dyrektywy, wówczas pierwszy wiersz powinien zawierać właściwe nazwy pól, w przeciwnym wypadku jako nazwy zostanie pobrany pierwszy wiersz danych.

Przykład użycia numerów pól w szablonie:

```
Section("", "Dokument")
{
    $klucz = [1]
    $dataDok = [2]
    $nip = [6]
}
```

Komentarze

Komentarzem w formacie SDF jest średnik umieszczony na początku linii.

Przykład:

```
;to jest komentarz. Następna linia również
"FV";1999-03-18; 1999-03-19;436576,56; "Holding S.A. ";233-22-33-234;01-223;"Warszawa"
```

Sekcje i podsekcje

W formacie SDF jedna linia tworzy jedną sekcję bez zagnieżdżonych podsekcji. Sekcja wejściowa SDF nie ma nazwy (nazwa jest pusta).

4.2.3 Format AM

Informacje ogólne

Format AM został wprowadzony w programie FK w wersji 2.0 i był używany do wymiany danych pomiędzy programami Sprzedaż oraz Finanse i Księgowość skończywszy na wersji 3.0. Jest to modyfikacja formatu SDF, w którym wprowadzono podział na sekcje oraz podsekcje. Format od wprowadzenia wersji FK 3.0 nie jest rozwijany, a bardzo mało elastyczna struktura tego formatu powoduje niemożność przyjmowania bardziej rozbudowanych dokumentów przy jego pomocy. Również szablony dla tego formatu nie są dalej rozwijane. Obecnie w rozwiązaniach Symfonii stosowany jest format AMS, zaś format AM pozostał na wypadek gdyby ktokolwiek chciał użyć archiwalnych wersji danych. Nie należy go używać do tworzenia nowych rozwiązań.

Pola danych

W formacie AM pola danych oddzielone są od siebie średnikami. Dodatkowo pola tekstowe ujęte są w cudzysłowie.

Przykład pól w formacie AM

```
"FV";1999-03-18; 1999-03-19;436576,56; "Holding S.A. "
```

Komentarze

Komentarzem w formacie AM jest średnik umieszczony na początku linii.

Przykład:

```
;to jest komentarz. Następna linia również  
;"DOK";1999-03-18; 1999-03-19;6576,56; "Holding S.A.";233-22-33-234;01-223; "Koło"
```

Sekcje i podsekcje

W formacie AM występują sekcje oraz podsekcje. Główną sekcją jest zawsze sekcja dokumentu, która zawiera podsekcje zapisów, rejestrów VAT oraz rozrachunków. Wyróżnikiem rodzaju sekcji jest zawartość pierwszego pola w wierszu danych:

„DOK” Sekcja dokumentu
„ZAP” podsekcja pierwszego zapisu w bilansującej się grupie (pozycji)
„ZAK” podsekcje kolejnych zapisów w bilansującej się grupie
„VAT” podsekcja rejestru VAT
„TRA” podsekcja nowej transakcji
„ROZ” podsekcja rozliczenia transakcji

4.2.4 Format CDF

Informacje ogólne

W formacie CDF poszczególne pola danych mają na stałe określony rozmiar (ilość znaków), jeśli aktualna wartość pola ma mniejszą liczbę znaków (np. cyfr w kwocie) wówczas reszta dopełniana jest spacjami do pełnego rozmiaru pola. W tym formacie nie ma separatorów pól.

Pola danych

Dla każdego pola w formacie CDF musi zostać określona szerokość tego pola (ilość znaków).

Szerokości poszczególnych pól podaje się bezpośrednio w specyfikacji formatu. Przykładowo jeśli linia danych zawiera 4 pola o szerokości odpowiednio 8, 4, 200 oraz 12 znaków, wówczas definicja tego formatu będzie wyglądać następująco:

```
#FORMAT „CDF,8,4,200,12”
```

Komentarze

W tym formacie nie ma komentarzy.

Sekcje i podsekcje

W tym formacie sekcję stanowi jeden wiersz danych. Nie ma możliwości tworzenia podsekcji, ani też sekcji różnego rodzaju.

4.2.5 Format AMS

Informacje ogólne

Format AMS jest obecnie powszechnie wykorzystywany w aplikacjach Symfonii. Stosuje on identyfikację każdej danej poprzez jej nazwę. Dzięki temu każda dana może być jednoznacznie zidentyfikowana niezależnie od tego, w którym miejscu się znajdzie, ani też jakie inne informacje i jakich miejscach są oprócz niej zawarte. Jedna linia danych w pliku zawiera tylko jedno pole danych. Nazwa pola danych jest oddzielona od jej wartości znakiem równości (podstawienia). Dane które są ze sobą logicznie połączone powinny zostać umieszczone w sekcji danych. Sekcja danych składa się z nazwy oraz znaków początku i końca sekcji. Wszystko co znajduje się pomiędzy znakami początku i końca sekcji traktowane jest jako zawartość sekcji. Sekcja danych jest elementarną porcją danych, przetwarzanych w całości przez procedury importu specjalnego. Szablony oparte na formacie AMS odpowiedzialne za wczytywanie konkretnych danych mogą przetwarzać dowolny podzbiór danych, zarówno w pojedynczej sekcji, jak i łącznie całych sekcji. Oznacza to że w pliku danych może być wiele nadmiarowych danych, które nie podlegają wczytywaniu i przetwarzaniu, oznacza to również że plik wejściowy może się zmieniać (może przybywać w nim kolejnych sekcji i kolejnych danych w sekcjach), a szablon pozostawiony w niezmienionej postaci nadal może te dane dokładnie tak samo przetwarzać. Format ten pozwala więc na elastyczną pracę z plikami danych.

Pola danych:

Każda dana (pole) składa się z nazwy danej oraz z jej wartości. Nazwa i wartość oddzielone są znakiem przypisania. Zarówno nazwa jak i wartość mogą zawierać dowolne znaki ASCII o kodach ≥ 32 . Spacje na początku i końcu nazwy pola oraz wartości są ignorowane.

Przykład:

```
Data dokumentu = 1997-09-12
Kwota = 12 000 245,56
Treść = Zakup towarów handlowych w firmie „PPHU Handel”
```

Nazwy pól danych mogą być dowolne, ale powinny być unikalne w ramach jednej sekcji. Jeśli pojawi się w jednej sekcji kilka pól o tej samej nazwie, wówczas dostępna będzie tylko wartość tylko pierwszego pola. Kolejne pola o powtórzonej nazwie będą niedostępne.

Pola danych mogą występować poza sekcjami tylko na samym początku pliku (tzw. dane globalne) mogą określać dodatkowo format pliku, w szczególności: format daty oraz separator dziesiętny w polach kwot, nadawcę pliku, datę pliku itp.

Przykład:

```
Data=rrrr-mm-dd
Separator01=,
Nadawca= SYSTEM SPRZEDAŻY v1.0
Data nadania=1997-09-12
```

Komentarze

Komentarze są to informacje przeznaczone wyłącznie dla osoby przeglądającej zawartość pliku, a nie dla programu wczytującego dane. Użycie komentarzy pozwala na przeprowadzanie eksperymentów w trakcie tworzenia szablonów importu (można szybko „usuwać” oraz „dodawać” dane).

Dwa znaki // (slash) oznaczają komentarz. Wszystko co znajduje się po tych znakach, aż do końca linii jest traktowane jak komentarz i jest ignorowane.

Przykład:

```
//ten tekst będzie zignorowany, to jest tylko komentarz
kwota = 12 000,89 //kwota brutto <- to też jest komentarz
//pole do testów = dane testowe
```

Sekcje i podsekcje

Sekcja składa się z nazwy sekcji, znaku początku sekcji, danych sekcji lub podsekcji, oraz znaku końca sekcji.

Nazwa sekcji może zawierać dowolne znaki ASCII o kodach ≥ 32 . Spacje na początku i końcu nazwy pola oraz wartości są ignorowane. Znakami początku i końca sekcji są nawiasy klamrowe.

UWAGA: W obecnej wersji importu specjalnego nazwa sekcji oraz znak początku sekcji muszą znajdować się w tej samej linii pliku wejściowego

Przykład:

```
Faktura VAT { //<- nazwa sekcji oraz początek sekcji
  data dokumentu = 1997-09-12 //<- dane sekcji
  kwota = 12 345,34
  kontrahent = ABACO HOLDING
} //koniec sekcji
```

Sekcje danych mogą być umieszczone wewnątrz innych sekcji danych, wówczas nazywamy je podsekcjami. Przykładem mogą być podsekcje zapisów w sekcji dokumentu

Przykład z zagnieżdżonymi sekcjami:

```
Faktura VAT { //<- nazwa sekcji oraz początek sekcji
  data dokumentu = 1997-09-12 //<- dane sekcji
  kwota = 12 345,34
  kontrahent { //<-nazwa podsekcji oraz początek podsekcji
nazwa = ABACO HOLDING
adres = Zalesie 12
  } //koniec podsekcji
} //koniec sekcji
```

Kolejność poszczególnych pól w sekcji nie ma znaczenia. Nazwy pól użyte w jednej sekcji mogą być powtórzone w dowolnej innej sekcji lub podsekcji.

Nazwy podsekcji są dowolne i w odróżnieniu od nazw pól mogą się powtarzać. Powtarzające się sekcje są dostępne i mogą być analizowane w kolejności ich występowania.

Każda sekcja danych powinna zawierać komplet danych niezbędnych do prawidłowego wprowadzenia danych do programu. W szczególności nie powinno być w nich bezpośrednich odwołań (relacji) do danych znajdujących się w innych plikach. Przykładowo sekcja dokumentu powinna zawierać podsekcję kontrahenta z podaniem wszystkich informacji o kontrahencie, a nie tylko numer kontrahenta w kartotece u nadawcy. Umożliwia to lepszą kontrolę wprowadzanych danych oraz gwarantuje spójność danych.

Mechanizmy importu specjalnego zastosowane w programie Symfonia Finanse i Księgowość zapewniają dużą elastyczność importu, dzięki zastosowaniu szablonów, które opisują sposób zamiany dowolnych danych wejściowych na wymaganą przez program postać. Mechanizm ten daje dużą swobodę w sposobie nazywania pól oraz sekcji danych a także formatu danych. Oczywiście dla każdego formatu danych wejściowych musi zostać stworzony oddzielny szablon opisujący ten format.

Przykładowe dane, generowane przez program Symfonia Płace, dane te mogą być wczytywane przy użyciu szablonu PLACE.AMS:

```
Dokument {
  NrKolejny =0
  Nazwa =PŁACE 02/1998
  Treść =Płatne gotówką luty 1998 otwarty
  DataDok =1997-11-14
  DataOper =1997-11-14
  DataWpływu =1997-11-14
  Kwota =1024.0000
```

```

Kontrahent =0
KontrahentFK =0
NIP =
Sygnatura =admin
SaldoPRK =0.0000
SaldoZRK =0.0000
KontoKasy =
NazwaKor =
DataKor =
ObsługujJak =DP
Zapis{
    ZAP/ZAK =P
    WN/MA =WN
    Kwota =192.8000
    Konto =100
    NumerDokDlaRK =numer = 0
    IdDlaRozl =1
    Opis =zaliczka na podatek
    zapisRownolegly =0
}
Zapis{
    ZAP/ZAK =K
    WN/MA =MA
    Kwota =203.8000
    Konto =210
    NumerDokDlaRK =numer = 0
    IdDlaRozl =2
    Opis =zaliczka na podatek i inne
    zapisRownolegly =0
}
Zapis{
    ZAP/ZAK =K
    WN/MA =WN
    Kwota =11.0000
    Konto =200-U6
    NumerDokDlaRK =numer = 0
    IdDlaRozl =5
    Opis =zaliczka na podatek
    zapisRownolegly =0
}
Pracownik {
    id =59
    nazwisko =Jankowski
    panienskie =
    imiel =Jerzy
    imie2 =Paweł
    imieOjca =Marian
    imieMatki =Alicja
    data urodzenia =1945-12-01
    miejsce urodzenia =Kobyłka

```

```

pesel =44120147891
nip =722-101-21-89
telefon =
Adres staly {
    miejscowosc =Kobyłka
    gmina =Kobyłka
    ulica =Kobielska
    numerDomu =12
    numerMieszk =7
    st_kod =12-876
    st_poczta =Rowy
    województwo =Opolskie
    kraj =
}
Adres tymczasowy {
    miejscowosc =Tłuszcz
    gmina =Kobyłka
    ulica =
    numerDomu =12
    numerMieszk =
    st_kod =12-300
    st_poczta =Kobyłka
    województwo =
    kraj =Polska
}
Rachunek {
    urzadzSkarbowy =Urząd skarbowy nr 3
    rach_symbol = PKO BP 0/12
    rach_numer =1342-3453-345345
}
}
Urzadz {
    id =6
    nazwa =Urząd Skarbowy Biała Podlaska
    ulica =Prosta
    nr_domu =19
    miejscowosc =Biała Podlaska
    kod =21-500
    Rachunek {
        rach_symbol =NBP o/o Biała Podlaska
        rach_numer =10101036-1531-222-3
    }
}
}
}

```

4.2.6 Format POLKA

Informacje ogólne

Historycznie jest to najstarszy format, stosowany przez program FK do wczytywania danych z programów innych producentów. Został opracowany na potrzeby współpracy z programem POLKA w wersji 2.0. Ze względu na upływ czasu i możliwość zmian przez producenta programu należy się liczyć z tym, że nie będzie mógł być wykorzystany z aktualnymi wersjami programu POLKA bez koniecznych modyfikacji. Jest

to format danych z nazwami pól, zgrupowanych w sekcje, które mogą być zagnieżdżane. Jest zbliżony do nowszego formatu AMS, który w pełni przejął jego funkcjonalność.

Sekcje i podsekcje

W formacie POLKA występują trzy rodzaje danych, każdy rodzaj jest inaczej zapisywany. Dane te są podzielone na trzy kategorie:

Dane globalne

Sekcja identyfikacja

Sekcje danych właściwych

Z punktu widzenia szablonów importu dane te tworzą trzy rodzaje sekcji wejściowych. Złożoność tego formatu wynika stąd, że każdy rodzaj sekcji i danych ma inną składnię.

Pola danych

Dane globalne są zbliżone do definicji w szablonie, czyli są to nazwy definicji poprzedzone znakiem # z następującą składnią:

```
#nazwa_definicji „wartość_definicji”
```

Sekcja IDENTYFIKACJA zawiera dane z nazwami pól, dla których obowiązującym separatorem nazwy i wartości jest znak dwukropka

```
Nadawca: "Polka™ 2.50"
```

Właściwe sekcje danych zawierają wartości z nazwami pól, a separatorem jest znak równości:

```
Nazwa klienta DO = HOLDING S.A.
```

W każdym z rodzajów danych w jednym wierszu znajduje się tylko jedna dana

Komentarze

4.2.7 Format SUBIEKT

Informacje ogólne

Jest to format stosowany przez wcześniejsze wersje programu SUBIEKT do eksportu danych z tego programu. Konstrukcja tego pliku jest w pewnym zakresie wzorowana na plikach typu INI. Jest to format, w którym dane wejściowe są grupowane w sekcje (nagłówki dokumentu) oraz podsekcje (pozycje towarowe). Format ten może nie odpowiadać potrzebom bieżącej wersji programu SUBIEKT i przedstawiony jest w celach poglądowych.

Pola danych

Pola danych są w formacie zbliżonym do SDF, czyli są to wartości oddzielone średnikami

Komentarze

W tym formacie nie ma komentarzy

Sekcje i podsekcje

W tym formacie nie ma jednorodnej definicji sekcji, występują zarówno jawnie nazwane sekcje zbliżone do formatu INI, jak też podsekcje będące wierszami w formacie SDF.

4.2.8 Format COMMA

Informacje ogólne

Format COMMA jest analogiczny do formatu SDF, z tą różnicą, że pola są od siebie oddzielone przecinkami.

Pola danych

W formacie COMMA pola danych oddzielone są od siebie przecinkami. Pola tekstowe mogą być ujęte w cudzysłowy, albo pozostać bez cudzysłowu. Jeśli nie używamy cudzysłowu, wówczas należy zadbać, aby wewnątrz pól tekstowych nie znalazł się znak przecinka, który zostanie potraktowany jako separator pól. Z tego względu bezpieczniej jest stosować teksty w cudzysłowie. Jeśli jakieś pole zawiera pustą daną (zero lub pusty tekst), można skrócić zapis takiej informacji, pisząc sam przecinek. Uwaga: w ten sposób nie można „skrócić” zapisu pierwszej danej w linii, gdyż tak linia zostałaby potraktowana jak komentarz.

Nazwy pól

Od wersji FK 4.0 można używać pierwszy wiersz jako nazwy pól. Służy temu dyrektywa:

```
#NAZWY_POL „TAK”
```

Oczywiście jeśli używamy tej dyrektywy, wówczas pierwszy wiersz powinien zawierać właściwe nazwy pól, w przeciwnym wypadku jako nazwy zostanie pobrany pierwszy wiersz danych.

Komentarze

Komentarzem w formacie SDF jest przecinek umieszczony na początku linii.

Sekcje i podsekcje

W formacie COMMA jedna linia tworzy jedną sekcję bez zagnieżdżonych podsekcji. Sekcja wejściowa nie ma nazwy (nazwa jest pusta).

4.2.9 Format Symfonia e-Faktury z EDS

Informacje ogólne

Format EDS jest hierarchicznym formatem XML (Maska plików danych *.XML) zawierającym dane wewnątrz tagów wejściowych zdefiniowanych w dokumencie XSD.

Pola danych

W formacie EDS pola danych zawarte są w notacji XML. Dane wejściowe mogą zawierać znaki specjalne np. **&**; **<**, które w procesie transformacji XSL zamieniane są na odpowiednie znaki w formacie ASCII. Dokument wejściowy może zawierać komentarze wewnątrz tagów **<!-- komentarz -->**

5 Elementy języka szablonów

5.1 Uwagi ogólne

Język szablonów składa się elementów znanych z innych języków programowania (np. polecenia, zmienne, podstawienia), oraz z elementów unikalnych, związanych wyłącznie z procesem przetwarzania danych. Polecenia mogą być zapisane dowolną wielkością liter, więc np. *While*, *WHILE*, *while* oznacza to samo. Poszczególne polecenia oddzielane są od siebie białymi znakami (spacja, tabulator, koniec linii).

5.2 Przetwarzanie danych przy pomocy szablonu

5.2.1 Wczytanie porcji danych wejściowych

Z pliku wejściowego dane są odczytywane sekwencyjnie, porcjami po jednej sekcji wejściowej. W zależności od formatu danych może to być jedna linia danych lub wiele linii tworzących całość informacji np. o jednym kontrahencie lub dokumencie. Dla każdej wczytanej sekcji danych identyfikowana jest jej nazwa. Dla formatów AM, AMS oraz POLKA nazwa sekcji jest podana w pliku danych, dla formatu SDF, CDF, COMMA nazwa sekcji wejściowej jest pusta (nie ma jej jawnie podanej w wierszu danych). Oczywiście oprócz identyfikacji nazwy są rozpoznawane wszystkie pola danych, oraz dokonywana jest konwersja kodów polskich liter. Tak przygotowane dane wejściowe są gotowe do przetwarzania przez szablon.

5.2.2 Identyfikacja procedury przetwarzania

Szablon składa się z jednej lub wielu procedur przetwarzających wczytane dane wejściowe. Dla każdej różnej sekcji wejściowej (dla każdej sekcji o innej nazwie) można utworzyć w szablonie oddzielną procedurę przetwarzającą te dane. Mechanizm ten umożliwia przetwarzanie z jednego pliku danych informacji różnego rodzaju. Przykładowo w jednym pliku mogą znajdować się dokumenty handlowe oraz kartoteki, oczywiście pod warunkiem, że będą znajdować się w sekcjach o różnych nazwach. Procedury przetwarzające w szablonie zaczynają się słowem SECTION i dwoma parametrami w nawiasie, po których następuje znak początku sekcji (otwierający nawias klamrowy), a kończą się znakiem końca sekcji (zamykający nawias klamrowy). Wszystko co znajduje się pomiędzy znakiem początku i końca sekcji jest zbiorem reguł przetwarzania w danej procedurze.

Przykład:

```
Section(„Faktura VAT”, „dokument”)
{
    $numer = [numer własny]
}
```

Po określeniu nazwy sekcji wejściowej jest odszukiwana w szablonie procedura, której pierwszy parametr jest taki sam jak nazwa sekcji wejściowej. Jeśli w szablonie nie ma takiej procedury, której pierwszy parametr odpowiada nazwie sekcji wejściowej, wówczas odszukiwana jest procedura, która ma pierwszy parametr pusty (puste cudzysłowy). Ta odszukana procedura będzie przetwarzała dane wczytanej sekcji wejściowej.

UWAGA: podczas poszukiwania właściwej procedury w szablonie ignorowana jest wielkość liter w nazwie sekcji, pomijane są białe znaki (spacje, tabulatory) z początku i końca nazwy, a białe znaki w środku nazwy sekcji są zamieniane na jedną spację.

5.2.3 Tworzenie sekcji wyjściowej

Po zidentyfikowaniu właściwej procedury w szablonie może zostać utworzona sekcja wyjściowa, która będzie zawierać dane przetworzone przez szablon, a które w konsekwencji zostaną wprowadzone do programu Finanse i Księgowość. Drugi parametr procedury SECTION określa nazwę sekcji wyjściowej. Jeśli nazwa ta nie zostanie podana, wówczas nie jest tworzona automatycznie sekcja wyjściowa (musi być utworzona jawnie np. poleceniem MakeSec). Program Finanse i Księgowość rozpoznaje i akceptuje następujące nazwy sekcji wyjściowych:

Dokument Odpowiada dokumentowi w buforze.

Zapis Odpowiada pojedynczemu zapisowi na dokumencie, a konkretnie jednej kwocie po jednej stronie konta. Może być użyta wyłącznie jako podsekcja sekcji *Dokument*.

Rejestr - Odpowiada pojedynczemu rejestrowi VAT w dokumencie. Może być użyta wyłącznie jako podsekcja sekcji *Dokument*.

Kontrahent Odpowiada kontrahentowi w kartotece kontrahentów stałych. Może występować samodzielnie lub jako podsekcja sekcji *Dokument*.

Incydentalny Odpowiada kontrahentowi w kartotece kontrahentów incydentalnych. Może występować samodzielnie lub jako podsekcja sekcji *Dokument*.

Pracownik - Odpowiada pracownikowi w kartotece pracowników. Może występować samodzielnie lub jako podsekcja sekcji *Dokument*.

Urząd Odpowiada urzędowi skarbowemu w kartotece urzędów.

Rachunek - Odpowiada rachunkowi bankowemu w kartotece rachunków bankowych.

Transakcja - Odpowiada nowej transakcji na dokumencie. Może być użyta wyłącznie jako podsekcja sekcji *Dokument* lub *Zapis*.

Rozliczenie - Odpowiada rozliczeniu transakcji na dokumencie. Może być użyta wyłącznie jako podsekcja sekcji *Dokument* lub *Zapis*.

5.2.4 Przetworzenie danych z sekcji wejściowej

Zawartość procedury w szablonie, czyli wszystko pomiędzy znakami początku i końca procedury jest zbiorem poleceń, które są wykonywane sekwencyjnie (polecenie po poleceniu) i powodują przetworzenie danych wejściowych, czyli w konsekwencji przepisaniu danych (po ewentualnej zmianie formatu, podzieleniu, połączeniu itp.) z sekcji wejściowej do sekcji wyjściowej. Podstawowym poleceniem przypisującym polu wyjściowemu wartość z pola wejściowego jest polecenie podstawienia.

Przykład:

```
$treść = [opis dokumentu]
```

W powyższym przykładzie polu wyjściowemu o nazwie *treść* przypisano wartość z pola wejściowego o nazwie *opis dokumentu*. Pola wyjściowe są zawsze identyfikowane przez znak \$ na początku nazwy pola. Pola wejściowe są zawsze identyfikowane przez podanie ich nazwy w nawiasach kwadratowych. Znakiem podstawienia jest znak równości (=) W przypadku nazw pól wejściowych nie ma znaczenia wielkość liter, którymi zapisana jest nazwa, ponadto białe znaki z początku i końca nazwy są ignorowane, a białe znaki w środku nazwy zamieniane są na jedną spację.

Dostęp do pól wejściowych jest swobodny, co oznacza że do wartości pól możemy odwoływać się w dowolnej kolejności (niezależnie od kolejności pól w danych wejściowych), oraz do jednego pola można odwoływać się wielokrotnie w danej procedurze. Do niektórych pól wejściowych (nieużywanych) możemy nie odwoływać się w ogóle.

Dostęp do pól wyjściowych jest również swobodny, co oznacza że do tych pól możemy odwoływać się w dowolnej kolejności i dowolną ilość razy, np. wielokrotnie modyfikować zawartość jednego pola. Do niektórych pól wyjściowych możemy nie odwoływać się w ogóle.

UWAGA: Możemy tworzyć dowolną ilość pól wyjściowych o dowolnych nazwach. Procedury importu nie sprawdzają i nie ostrzegają jeśli utworzone pole nie ma swojego odpowiednika w bazach danych programu. Przykładowo w sekcji *Kontrahent* możemy utworzyć pole *Zawód*, ale pole to zostanie zignorowane, gdyż w kartotece kontrahentów w programie *Finanse i Księgowość* nie ma pola o nazwie *Zawód*. Wszystkie rozpoznawalne nazwy pól poszczególnych sekcji wyjściowych, wraz z regułami ich wypełniania są opisane w dalszej części dokumentu.

5.2.5 Podsekcje w danych wejściowych

W sekcji wejściowej może znajdować się dowolna liczba podsekcji. Przykładowo w sekcji dokumentu mogą znajdować się podsekcje zapisów, rejestrów VAT oraz rozrachunków. Aby móc przetwarzać dane również w podsekcjach, należy w szablonie, wewnątrz procedury przetwarzającej sekcję wejściową umieścić odpowiednie podprocedury.

Podprocedury mają dokładnie taką samą składnię jak procedury, a umieszczane są wewnątrz procedur (stąd ich nazwa).

Przykład:

```
Section(„Faktura VAT”, „dokument”)
{
    $numer = [numer własny]
    Section(„Odbiorca”, „Kontrahent”)
    {
        $nazwa = [nazwa odbiorcy]
    }
}
```

W powyższym przykładzie w procedurze obsługującej sekcję wejściową o nazwie *Faktura VAT* znajduje się podprocedura obsługująca podsekcję danych wejściowych o nazwie *Odbiorca*.

Jeśli w trakcie wykonywania kolejnych poleceń procedury zostanie zidentyfikowana podprocedura, wówczas wykonywane są kolejno następujące kroki:

w danych wejściowych zostanie odszukana podsekcja o nazwie takiej, jak pierwszy argument podprocedury (w naszym przykładzie będzie to podsekcja *Odbiorca*). Jeśli nie odnajdziemy podsekcji wejściowej o takiej nazwie, wówczas generowany jest błąd wykonania. Jeśli pierwszy argument podprocedury będzie pusty, wówczas w sekcji wejściowej jest odnajdywana pierwsza w kolejności podsekcja. Wszystkie pola znalezionej podsekcji stają się dostępne poleceniom podprocedury w szablonie.

Drugi parametr podprocedury określa nazwę podsekcji wyjściowej, która ma zostać utworzona. Jeśli nazwa ta nie zostanie podana, wówczas nie jest tworzona automatycznie sekcja wyjściowa (może być utworzona jawnie np. poleceniem *MakeSec*). Utworzona w ten sposób podsekcja wyjściowa jest umieszczana wewnątrz sekcji wyjściowej (tworząc analogiczną strukturę jak sekcje i podsekcje wyjściowe)

Wszystkie polecenia podprocedury w szablonie są wykonywane sekwencyjnie, analogicznie jak polecenia głównej procedury.

W trakcie wykonywania poleceń podprocedury, dostępne są wyłącznie pola z odnalezionej podsekcji wejściowej. W szczególności nie można odwołać się bezpośrednio do pól głównej sekcji wejściowej. (jakkolwiek jest możliwe przekazanie wartości wybranych pól poprzez zmienne szablonu).

Wszystkie zmienne wyjściowe (pola sekcji wyjściowej) są umieszczane w nowoutworzonej podsekcji wyjściowej. Istnieje możliwość odwołania się bezpośrednio do pól głównej sekcji wyjściowej, w tym celu nazwę pola wyjściowego należy poprzedzić podwójnym znakiem \$, np. \$\$nazwa.

Jeśli nie podamy nazwy podsekcji wyjściowej, wówczas podsekcja ta nie zostaje utworzona i wówczas wszystkie pola wyjściowe są tworzone w głównej (bieżącej) sekcji wyjściowej.

Po zakończeniu wykonywania poleceń podprocedury następuje zamknięcie podsekcji danych wejściowych (staje się ona już niedostępna dla szablonu), oraz zamknięcie sekcji wyjściowej (ona również staje się niedostępna z szablonu). Jeśli po końcu podprocedury znajdują się inne polecenia procedury głównej, są one dalej wykonywane sekwencyjnie, aż do końca procedury. Dla tych poleceń znów dostępne są wszystkie pola sekcji wejściowej, oraz wszystkie pola sekcji wyjściowej.

UWAGA: Podprocedury mogą być wielokrotnie „zagnieżdżane”, to znaczy umieszczane jedna w drugiej, stosownie do zagnieżdżania kolejnych podsekcji w danych wejściowych. Przykładowo w sekcji *Dokument* może być umieszczona podsekcja *Zapis*, a w niej następna podsekcja o nazwie

Transakcja, odpowiadająca nowej transakcji dla tego zapisu. W szablonie musimy wówczas utworzyć procedurę do obsługi sekcji *Dokument*, w niej umieścić podprocedurę do obsługi podsekcji *Zapis*, a w niej z kolei podprocedurę do obsługi podsekcji *Transakcja*.

Zasady wykonywania kolejnych zagnieżdżonych podprocedur są identyczne jak pierwszej podprocedury.

5.2.6 Przetwarzanie wielu sekcji o tej samej nazwie

Import specjalny umożliwia analizowanie wielu podsekcji o identycznej nazwie, umieszczonych w jednej sekcji wejściowej. Przykładem mogą być podsekcje *Zapis* w sekcji *Dokument*. W przypadku raportu kasowego może być ich nawet kilkaset w jednym dokumencie.

Przetwarzanie powtarzających się podsekcji może być przeprowadzone na dwa sposoby:

Dodanie trzeciego parametru podprocedury o wartości *REPEAT* (powtarzaj), powoduje cykliczne odszukanie i przetworzenie wszystkich podsekcji wejściowych o nazwie podanej jako pierwszy parametr podprocedury. Dopiero po przetworzeniu wszystkich podsekcji wejściowych sterowanie przekazywane jest do poleceń umieszczonych po znaku końca podprocedury.

Umieszczenie podprocedury wewnątrz pętli programowej *While*, w tym wypadku podprocedura zostanie wykonana tyle razy, ile zostanie określone w warunku sterującym pętlą. Aby przetworzyć wszystkie podsekcje wejściowe, w warunku sterującym pętli *While* można umieścić polecenie sprawdzające, czy istnieje następna podsekcja wejściowa.

5.3 Komentarze

Komentarze mogą być jednoliniowe lub wieloliniowe (blokowe)

Jednoliniowe: są to dwa znak // (slash). Wszystko co znajduje się po tych znakach, aż do końca linii jest ignorowane.

Wieloliniowe: początek komentarza stanowią dwa znaki: /* umieszczone na początku linii. Koniec komentarza stanowią dwa znaki */ umieszczone na początku jednej z następnych linii. Wszystko co znajduje się pomiędzy znakami początku i końca komentarza jest ignorowane

5.4 Definicje

Definicje służą do określenia stałych parametrów początkowych wczytywania danych, np. ustalenie formatu danych itp.

Składnia definicji jest następująca:

```
#nazwa_definicji „wartość definicji”
```

Definicje tworzone dynamicznie podczas wykonywania kodu szablonu (poleceniem DEF) mogą być używane jako zmienne globalne, które nie znikają po zakończeniu analizy jednej sekcji i są "widziane" dla wszystkich sekcji wejściowych i wszystkich podsekcji w jednym pliku danych.

5.5 Polecenia globalne

Polecenia globalne wykonywane są jeden raz przed lub po wykonaniu właściwego kodu szablonu. Umożliwiają np. wczytywanie danych z wielu plików. Składnia jest następująca :

```
#nazwa_polecenia( „argument polecenia”, „argument polecenia” ....)
```

5.6 Polecenia sekcji

Polecenia (funkcje wbudowane) umożliwiają przetwarzanie zawartości danych wejściowych, zmiennych, pól oraz wartości bezpośrednich. Argumenty i wartość zwracana przez funkcje jest zawsze tekstowa. Argumentem może być pole, zmienna, lub inna funkcja. Argumenty są umieszczone w nawiasach okrągłych, poszczególne argumenty są oddzielane przecinakami. Separatorem poleceń jest biały znak (spacja, tabulator, koniec linii).

Nazwa funkcji zawsze zaczyna się literą, po której następują litery lub cyfry. W szablonie występują wyłącznie funkcje wbudowane, nie ma możliwości tworzenia własnych funkcji (podprogramów) wywoływanych z różnych sekcji szablonu. Poszczególne funkcje zostaną opisane szczegółowo w dalszej części opracowania. Polecenia sekcji mogą być użyte wyłącznie wewnątrz sekcji. Użycie ich poza sekcją jest traktowane jako błąd.

5.7 Sekcje szablonu

Wszystkie polecenia szablonu poza definicjami i poleceniami globalnymi muszą być umieszczone w jakiejś sekcji. Każdy rodzaj sekcji wejściowej musi mieć swój odpowiednik w sekcji szablonu, który stanowi zbiór zasad interpretacji danych wejściowych.

5.8 Pola wejściowe

Pola wejściowe reprezentują wczytane dane sekcji. Mogą być argumentami podstawienia oraz argumentami dla funkcji szablonu. Pola są zawsze tekstowe (zawierają dane tekstowe) niezależnie od tego, czy zawierają tekst czy datę lub liczbę. Aby pobrać pole należy podać nazwę pola w nawiasach kwadratowych. Nazwy pól nie mogą przekraczać 60 znaków, a zawartość pola 300 znaków. Kolejność użycia pól wejściowych jest dowolna.

Przykład:

```
[Stawka VAT]
```

```
[1]
```

Uwaga: spacje z początku i końca nazwy są ignorowane, a białe znaki w środku normalizowane do jednej spacji [Stawka VAT] to jest to samo co [Stawka VAT] Pola wejściowe bez nazwy podanej explicite (np. dla formatu SDF) są automatycznie nazywane kolejnymi numerami odpowiadającymi kolejności w pliku danych: [1], [2], [3] itp.

5.9 Zmienne

Zmienne są tworzone automatycznie w momencie pierwszego pojawienia się ich nazwy po lewej stronie znaku podstawienia. Nie trzeba deklarować zmiennych. Zmienne są wyłącznie tekstowe. Zmienne zawsze związane są z aktualnie utworzoną sekcją wyjściową i po przetworzeniu całej sekcji są usuwane. Długość nazwy zmiennej nie może przekraczać 32 znaków, z wartością zmiennej 300 znaków.

Zmienne lokalne: widziane są tylko w ramach bieżącej sekcji wyjściowej i po jej zamknięciu są niedostępne. Zmienne lokalne zaczynają się znakiem @ po którym następuje dowolny ciąg liter, cyfr oraz znaków podkreślenia. Np. @zmienna, @rex1

Zmienne globalne: widziane są w głównej sekcji wyjściowej oraz we wszystkich jej podsekcjach. Zmienne globalne zaczynają się dwoma znakami @@ po których następuje dowolny ciąg liter, cyfr oraz znaków podkreślenia. Np. @@zmienna @@rex1. Każda zmienna lokalna utworzona w głównej sekcji wyjściowej jest widziana jako zmienna globalna w podsekcjach wyjściowych.

Istnieje zmienna specjalnego przeznaczenia o nazwie @autoinsert

5.10 Pola wyjściowe

Pola sekcji wyjściowej są tworzone automatycznie w momencie pierwszego pojawienia się ich nazwy po lewej stronie znaku podstawienia. Nie trzeba deklarować pól wyjściowych. Pola wyjściowe są wyłącznie tekstowe. Pola wyjściowe zawsze związane są z aktualnie utworzoną sekcją wyjściową i po przetworzeniu całej sekcji są niedostępne. Długość nazwy pola nie może przekraczać 60 znaków, z wartością pola 300 znaków. Nazwą pola może być dowolny ciąg liter, cyfr oraz znaków podkreślenia. W przypadku gdy w nazwie pola muszą wystąpić inne znaki, wówczas należy nazwę pola ująć dodatkowo w nawiasy okrągłe

Pola bieżącej sekcji wyjściowej: widziane są tylko w ramach bieżącej sekcji wyjściowej i po jej zamknięciu są niedostępne. Zaczynają się znakiem \$, np. \$pole \$(data dokumentu)

Pola głównej sekcji wyjściowej: Aby mieć dostęp do pól głównej sekcji wyjściowej z dowolnej podsekcji należy nazwę pola poprzedzić dwoma znakami \$\$. Przykładowo: \$\$\$(data dokumentu)

5.11 Argumenty bezpośrednie

Argumenty bezpośrednie są wyłącznie tekstowe. Argumenty muszą być umieszczone w cudzysłowie. Wyjątek stanowią liczby, które mogą być podane bez cudzysłowu (choć one również są przetwarzane do postaci tekstowej) Znak \ (backslash) w tekście oznacza zignorowanie specjalnego znaczenia kolejnego znaku, a w szczególności znaku cudzysłowu " i samego znaku \

5.12 Opis szablonów XSL

Szablony transformacji:

- efaktury.xslt – import za pomocą profilu EDS
- dokumenty.xslt – import dokumentów z formatki (DIM, FVZ) dokumentów niewalutowych i walutowych bez dekretacji różnic kursowych wynikających z Kurs CIPIT i Kurs VAT

- `dokumnety_walutowe.xslt` – import dokumentów z formatki (FWZ) dokumentów walutowych z wykorzystaniem różnic Kurs CITPIT i Kurs VAT

Import za pomocą profilu EDS polega na transformacji e-Faktury w formacie XML do dokumentu tekstowego bazującego na formacie 3. W tym przypadku szablon XSL określony jest w parametrach profilu. Transformacja XSL jest pierwszym etapem przetwarzania. Efekt przetwarzania importowany jest w kolejnym kroku za pomocą szablonu AMS określonego w parametrach profilu.

Celem szablonu XSLT jest zdefiniowanie mapowania pól z dokumentu wejściowego z EDS do pól Formatu 3.

Przykład:

Pole wejściowe **GUID** przepisuj do **eguid**.

```
eguid =<xsl:value-of
select="/FormatXML/DokumentHandlowy[position()=1]/Odbiorca/GUID"/>
```

Przykład:

Sekcję **Wystawca** z sekcji **DokumentHandlowy** przepisuj do sekcji **Dane sprzedawcy** w sekcji **Dokument**.

```
<xsl:template match="DokumentHandlowy/Wystawca">&#10; Dane sprzedawcy{
    khguid =<xsl:value-of select="GUIDEDS"/>
    nazwa =<xsl:value-of select="Nazwa"/>
    ...
}
</xsl:template>
```

Szczegółowy opis formatu XSLT znajduje się na stronach W3C - <http://www.w3.org/TR/xslt>

Szczegółowy opis języka XPath w wersji 1.0 - <http://www.w3.org/TR/xpath/>

6 Spis definicji

#DATA „format daty”

Specyfikacja formatu daty, używana przez polecenie DATE. Format podajemy przez określenie położenia i separatorów poszczególnych segmentów daty. Poszczególne symbole oznaczają: R - rok, M - miesiąc, D - dzień. Nie jest konieczne specyfikowanie długości poszczególnych segmentów, tak więc RRRR-MM-DD jest równoważne R-M-D. Jeśli numer roku zawiera tylko dwie ostatnie cyfry, nastąpi automatyczne uzupełnienie do 4 cyfr, wg. następującego schematu:

Jeśli rok < 70 dodajemy 2000 (lata 2000-2069)

Jeśli rok >= 70 ale rok < 100 dodajemy 1900 (lata 1970 – 1999)

Wewnętrznym formatem daty w szablonie jest format RRRR-MM-DD. Jeśli w pliku danych daty są kodowane w ten sposób, wówczas mogą być podstawiane do pól wyjściowych bezpośrednio, w każdym innym przypadku należy dokonać zamiany formatu poleceniem DATE.

Przykład:

```
#DATA „RR-MM-DD” //2 cyfry roku, miesiąc, dzień. Separatorem jest myślnik
```

#SEPARATOR_01 „separator dziesiętny”

Specyfikacja separatora dziesiętnego w liczbie, używana przez polecenie MONEY. Określa znak, który oddziela część całkowitą i ułamkową w liczbie. Wszystkie znaki różne od +, -, cyfr i separatora dziesiętnego są ignorowane bez żadnego ostrzeżenia. W konsekwencji nie jest konieczne specyfikowanie stosowanego separatora tysięcznego. Domyślnym separatorem dziesiętnym dla szablonu jest kropka, oraz nie ma separatora tysięcznego. Jeśli w pliku danych kwoty są kodowane w ten sposób, wówczas mogą być podstawiane do pól wyjściowych oraz poleceń arytmetycznych bezpośrednio, w każdym innym przypadku należy dokonać zamiany formatu poleceniem MONEY.

Przykład:

```
#SEPARATOR_01 " ," //separator przecinek
```

#FORMAT „nazwa formatu”

Specyfikacja formatu danych wejściowych. W każdym szablonie musi być podana nazwa formatu, gdyż steruje ona wczytywaniem i interpretacją danych wejściowych. Predefiniowane formaty są następujące:

SDF - Format pól oddzielonych średnikami. Jedna linia pliku stanowi jedną sekcję danych

CDF,a,b,c... - Format pól o stałej długości. Jedna linia pliku stanowi jedną sekcję danych. Wartości a,b,c itd. Określają szerokości (w znakach) kolejnych pól danych.

COMMA - Format pól oddzielonych średnikami. Jedna linia pliku stanowi jedną sekcję danych

POLKA Format - pliku używany do wczytywania danych z programu POLKA do programu Symfonia Finanse i Księgowość.

SUBIEKT - Format stosowany przez program SUBIEKT firmy Insert.

AM - Format wymiany danych używany w wersji 2.0 programu Finanse i Księgowość (import danych z innych modułów Symfonii).

AMS - Nowy, zalecany format wymiany danych używany od wersji 3.0 programu Finanse i Księgowość (używany obecnie przez większość programów Symfonii – Kadry, Płace, Handel, Faktura, Środki Trwałe i inne).

Ilość predefiniowanych formatów może ulegać zmianie w kolejnych wersjach programu.

Istnieje ponadto możliwość dodawania własnych definicji formatów danych.

#TYP_KODOWANIA „symbol sposobu kodowania”

Polecenie określa sposób kodowania polskich znaków w pliku wejściowym. Każda wczytywane informacja (nazwa sekcji lub danej oraz wartość danej) jest translowana zgodnie z podanym sposobem kodowania. Predefiniowane sposoby kodowania są następujące:

MAZOVIA

LATIN2

DHN

PN_ISO

#TABLICA_KODOWANIA „tablica kodowania”

Polecenie definiuje w sposób jawny sposób przekodowania polskich liter. Tablica przekodowania składa się z kolejnych par liter lub kodów liter oddzielonych przecinkami (każdy z elementów pary może być podany dowolnie jako znak lub jako liczba określająca kod znaku). Pierwszy znak z pary określa kod litery w pliku wejściowym, a drugi znak określa kod po translacji.

Przykład:

```
#TABLICA_KODOWANIA „165,a,164,A,134,ć,143,Ć” //fragment kodowania LATIN2
```

7 Spis operatorów

7.1 Podstawienie

Operator podstawienia (znak równości) umożliwia nadanie wartości zmiennej lub polu wyjściowemu składnia

Zmienna lub pole = wartość, pole, zmienna lub funkcja

Nie można podstawiać wartości do pól wejściowych, Jako argument nie może być podana nazwa zmiennej lub pola nie użytego jeszcze po lewej stronie podstawienia (nie utworzonego).

Przykład:

```
$kwota = [kwota netto 7%]
```

7.2 Negacja

Operator negacji (znak wykrzyknika) neguje wartość logiczną argumentu. Analogiczną funkcję spełnia polecenie NOT.

Gdy argument nie jest pustym tekstem zwraca pusty tekst

Gdy argument jest pustym tekstem zwraca tekst z jedną spacją

Przykład:

```
IF( ! [numer dokumentu] )  
  ERROR( „Numer dokumentu nie może być pusty!”)
```

8 Spis poleceń

8.1 Polecenia globalne

#EXECUTE(nazwa pliku, nazwa szablonu)

Polecenie powoduje wczytanie danych z podanego pliku, według podanego szablonu. Dopiero po zakończeniu wczytywania podanego pliku nastąpi wczytanie danych z bieżącego pliku wejściowego. Polecenie umożliwia import danych kolejno z kilku plików (np. w przypadku osobnych plików danych kontrahentów, dokumentów itp.)

#RETURN(format_danych_zwrotnych, nazwa_pliku, rodza_danych)

Polecenie specyfikuje format pliku informacji zwrotnych odsyłanych do nadawcy danych, nazwę tego pliku oraz określa rodzaj tych informacji. Informacje zwrotne odsyłane są po zakończeniu importu.

Istnieją dwie zasadnicze grupy informacji zwrotnych: informacje o uzgodnieniach, oraz informacje transakcjach.

Informacje o uzgodnieniach

Informacje o uzgodnieniach mogą być wysłane tylko w formacie AMZ. Argument *rodzaj_danych* określa, jakie informacje zostaną odesłane. Jest to tekst zawierający dowolną kombinację podanych poniżej liter:

K	połączenia kartoteki kontrahentów
P	połączenia kartoteki pracowników
U	połączenia kartoteki urzędów
R	połączenia kartoteki rachunków bankowych
J	połączenia kartoteki kontrahentów incydentalnych
D	połączenia typów dokumentów
V	połączenia definicji rejestrów VAT

Odesłaniu podlegają informacje o połączeniach dokonanych od ostatniego wysłania informacji zwrotnych. Istnieje też możliwość ponownego wysłania dowolnych poprzednich informacji zwrotnych,

Informacje o transakcjach

Możliwość przesyłania do nadawcy informacji o transakcjach z kontrahentami, którzy zostali zaimportowani od danego nadawcy została wprowadzona w wersji FK 4.0c.

Możliwy jest wybór trzech poziomów szczegółowości danych:

1. Tylko informacje o sumie nierozliczonych należności i zobowiązań z danym kontrahentem, wówczas w polu *rodzaj_danych* należy wpisać „Rozrachunki”
2. Dodatkowo oprócz informacji o sumie nierozliczonych należności i zobowiązań z danym kontrahentem zwracana jest lista wszystkich transakcji z zaznaczeniem kwoty do rozliczenia, terminu płatności oraz dokumentu źródłowego, wówczas w polu *rodzaj_danych* należy wpisać „Rozrachunki T”
3. Dodatkowo oprócz informacji o poszczególnych transakcjach zwracane są wszystkie częściowe rozliczenia transakcji, wówczas w polu *rodzaj_danych* należy wpisać „Rozrachunki TR”

Dla wariantu 2 oraz 3 można dodatkowo określić, czy zwracana ma być informacja o wszystkich transakcjach z danym kontrahentem, czy tylko informacja o transakcjach nie rozliczonych. Domyślnie zwracana jest informacja o wszystkich transakcjach. Aby wybrać tylko transakcje nie rozliczone, należy w specyfikacji dodać literę N, np.

„Rozrachunki TN” – tylko transakcje nierozliczone

„Rozrachunki TRN” – transakcje nierozliczone i ich rozliczenia częściowe

Dane zwrotne o rozrachunkach mogą być wysłane w formacie AM (jest to zalecany format) lub w formacie AMR - pola oddzielone średnikami, zawiera 3 rodzaje linii danych dla trzech możliwych poziomów szczegółowości.

8.2 Polecenia sekcji

CAT(arg1, arg2, ...)

Polecenie łączy teksty – do 30 argumentów. Długość tekstu wynikowego jest ograniczona do 16 000 znaków. Obcięcie dłuższego tekstu nie jest sygnalizowane.

Przykład:

```
@(numer ewidencyjny) = CAT([typ dokumentu], „ - ”, [numer dokumentu])
```

SUMx(arg1, arg2, ...)

Polecenie oblicza sumę argumentów (do 30 argumentów).

X oznacza dokładność obliczeń (zaokrąglenie wyniku, nie składników). Obecnie jest to:

0 – całkowite

2 – do grosza

6 – do 6 miejsc po przecinku (np. dla kursu walut itp.)

Przykład:

```
$brutto = SUM2( [netto], [vat])
```

MULx(arg1, arg2, ...)

Polecenie oblicza iloczyn argumentów (do 30 argumentów).

X oznacza dokładność obliczeń (zaokrąglenie wyniku, nie składników). Obecnie jest to:

0 – całkowite

2 – do grosza

6 – do 6 miejsc po przecinku (np. dla kursu walut itp.)

Przykład:

```
$vat = MUL2( [netto], [stawka vat w %], 0.01)
```

DIVx(arg1, arg2, ...)

Polecenie oblicza wynik dzielenia pierwszego argumentu przez wszystkie kolejne argumenty (do 30 argumentów).

X oznacza dokładność obliczeń (zaokrąglenie wyniku, nie składników). Obecnie jest to:

6 – do 6 miejsc po przecinku (np. dla kursu walut)

Przykład:

```
$kurs = DIV6( [kwota zł], [kwota wal])
```

NOT(arg)

Polecenie neguje wartość logiczną argumentu, analogicznie jak operator negacji.

Gdy argument nie jest pustym tekstem zwraca pusty tekst

Gdy argument jest pustym tekstem zwraca tekst z jedną spacją

Przykład:

```
IF( NOT( [numer dokumentu] ) )  
  ERROR( „Numer dokumentu nie może być pusty!”)
```

EQUAL(arg1, arg2 [, sposób])

Porównuje dwa argumenty tekstowe.

Gdy argumenty nie są równe polecenie zwraca pusty tekst

Gdy argumenty są równe polecenie zwraca tekst z 1 spacją
Polecenie może zawierać opcjonalnie sposób porównania tekstów w postaci tablicy translacji, która składa się kolejnych par liter lub kodów liter oddzielonych przecinkami. Pierwszy znak z pary określa kod litery w pliku wejściowym, a drugi znak określa kod po translacji. Polecenie umożliwia również porównanie bez rozróżniania małych i dużych liter, w tym celu należy jako tablicę translacji podać jedną literę L.

Przykład został podany przy opisie polecenia IF

TRANSLATE(wartość, tablica translacji)

Polecenie dokonuje translacji (zamiany liter w tekście) według podanego wzorca (tablicy translacji).

Tablica translacji składa się kolejnych par liter lub kodów liter oddzielonych przecinkami. Pierwszy znak z pary określa kod litery w pliku wejściowym, a drugi znak określa kod po translacji. Polecenie umożliwia również zamianę małych liter na duże lub odwrotnie, w tym celu należy jako tablicę translacji podać jedną literę:

L – zamiana dużych liter na małe (Lower case)

U – zamiana małych liter na duże (Upper case)

Przykład:

```
@opis = TRANSLATE([opis dokumentu], „165,a,164,A,134,ć,143,Ć”) //fragment kodowania LATIN2
```

WARNING(arg1, arg2,)

Wprowadza do okna importu komunikat o statusie ostrzeżenia. Polecenie nie przerywa importu danych. Argumenty są łączone w jedną linię komunikatu. Może być użytych do 30 argumentów.

Przykład:

```
WARNING(„Dokument: ” , [nazwa dokumentu], „ posiada błędy -> został zignorowany”)
```

INFO(arg1, arg2, ...)

Wprowadza do okna importu komunikat o statusie informacji. Polecenie nie przerywa importu danych. Argumenty są łączone w jedną linię komunikatu. Może być użytych do 30 argumentów.

Przykład:

```
INFO(„Dokument: ” , [nazwa dokumentu], „ został zapisany”)
```

ERROR(arg1, arg2, ...)

Wprowadza do okna importu komunikat o statusie błędu. Polecenie przerywa wykonanie importu. Argumenty są łączone w jedną linię komunikatu. Może być użytych do 30 argumentów.

Przykład:

```
ERROR(„Nieznana sekcja wejściowa: ”, Name() )
```

ASK(arg1, arg2, ...)

Umożliwia wyświetlenie okna dialogowego komunikatu (MessageBox). Argumenty są sklejane w jedną linię komunikatu . Może być użytych do 30 argumentów. Można sterować ilością oraz tekstem na przyciskach poleceń. Obecnie można umieścić do 3 przycisków oraz jedną bitmapę. Sposób sterowania przyciskami może być różny w zależności od programu importującego (może też nie być obsługiwany w ogóle). Dla programu Finanse i Księgowość składnia jest następująca:

„Treść komunikatu DefBtn: nazwa przycisku = wartość [Btn: nazwa przycisku = wartość] [Ico: symbol]”

Oznaczenia:

DefBtn: - oznacza przycisk wybierany automatycznie po naciśnięciu klawisza ENTER (domyślny). Może wystąpić tylko jeden raz

Btn: - oznacza pozostałe przyciski. Łącznie mogą wystąpić tylko trzy przyciski

Nazwa przycisku - jest to dowolny tekst umieszczony na przycisku. Litera poprzedzona znakiem & będzie podkreślona i stanowi akcelerator, działający analogicznie jak w innych oknach dialogowych.

Wartość - jest to wartość zwracana po wybraniu przycisku. Jest to dowolna liczba z zakresu 1..1000. Użycie wartości=2 oznacza możliwość zamknięcia dialogu klawiszem Escape

Ico: - oznacza ikonę z lewej strony komunikatu

Symbol - symbol oznaczający ikonę: S - ikona (STOP), ?- ikona (?), ! – ikona (!)

Przykład:

```
@action = ASK(„Dokument posiada błędy, czy go zapisać? DefBtn:&Zapisz=3  
Btn:&Anuluj=2 Ico:?”)
```

IF(warunek) polecenie lub blok poleceń [ELSE polecenie lub blok poleceń]

Polecenie umożliwia sprawdzenie wartości wyrażenia i w zależności od wyniku sprawdzenia wykonanie (lub nie) określonych poleceń. Jako warunek może być podane dowolne wyrażenie, pole, zmienna lub funkcja. Jeśli obliczone wyrażenie jest niepuste (niepusty tekst), wówczas wykonywane jest polecenie znajdujące się bezpośrednio za IF. Jeśli obliczone wyrażenie jest puste (pusty tekst), wówczas wykonywane jest polecenie znajdujące się za słowem ELSE, o ile słowo to występuje. Jeśli wykonane ma być więcej niż jedno polecenie (blok poleceń) należy wszystkie te polecenia umieścić w nawiasach klamrowych.

Przykład :

```
IF( NOT( EQUAL( [ typ kontrahenta], „Odbiorca”) ) )  
    $dostawca = „TAK”  
ELSE  
{  
    $odbiorca = „TAK”  
}
```

WHILE(wyrażenie) polecenie lub blok poleceń

Polecenie umożliwia wielokrotne wykonanie polecenia lub bloku poleceń, dopóki obliczone wyrażenie jest niepuste. Jeśli wykonane ma być więcej niż jedno polecenie (blok poleceń) należy wszystkie te polecenia umieścić w nawiasach klamrowych.

Przykład:

```
@cnt = 0  
WHILE( NOT( EQUAL( @cnt, 100) ) ) //pętla wykona się 100 razy  
{  
    @cnt = SUM0( @cnt, 1)  
}
```

REX(dana we, wyrażenie regularne [, polecenie do wykonania po dopasowaniu])

Polecenie dopasowuje wyrażenie regularne do podanego tekstu. Tworzy i ustawia zmienne lokalne o nazwach @rex0..@rex9 odpowiadającą dopasowanemu tekstowi oraz zaznaczonym sekcjom wyrażenia regularnego (szczegółowy opis wyrażeń regularnych znajduje się w pliku pomocy do języka raportów). Jeśli zostanie dopasowane wyrażenie, wówczas (i tylko wówczas) obliczana i zwracana przez funkcję jest wartość trzeciego wyrażenia (o ile istnieje). Jeśli trzeci argument nie istnieje, wówczas zwracane jest całe dopasowane wyrażenie.

Dana we – tekst w którym poszukujemy dopasowania

Wyrażenie regularne – specyfikacja wyszukiwanego tekstu

Przykład:

```
//Odwrocenie formatu daty z RRRR-MM-DD do DD-MM-RRRR  
$data = REX(„1997-07-12”, „{????}{/-?}/{-??}”, CAT(@rex3, @rex2, „-” , @rex1))
```

EXIST(nazwa)

Polecenie sprawdza, czy w bieżącej sekcji wejściowej znajduje się pole lub podsekcja o podanej nazwie. Jeśli pole lub podsekcja istnieje, polecenie zwraca tekst zawierający jedną spację. W przeciwnym wypadku polecenie zwraca pusty tekst.

Przykład:

```
IF( EXIST( „Rejestr” ) )  
  INFO(„Znaleziono pole lub sekcję o nazwie 'Rejestr' ”)
```

FIELD(nazwa)

Polecenie zwraca wartość pola o podanej nazwie. W odróżnieniu od pobrania wartości przez podanie nazwy w nawiasach kwadratowych, argumentem tego polecenia może być dowolne wyrażenie.

Przykład:

```
//wysłanie do okna komunikatów nazw i wartości wszystkich pól z sekcji w  
formacie SDF  
@name = 1  
WHILE EXIST( @name )  
{  
  INFO( @name, " = ", FIELD( @name ) )  
  @name = SUM( @name, 1 )  
}
```

NAME()

Polecenie zwraca nazwę bieżącej sekcji (lub podsekcji) wejściowej.

DATE([wartość])

Polecenie dokonuje zamiany daty z formatu pliku wejściowego (określonego poleceniem #DATA), do postaci RRRR-MM-DD, czyli 4 cyfry roku - 2 cyfry miesiąca - 2 cyfry dnia.

Jeśli argument nie zostanie podany, wówczas polecenie zwróci aktualną datę systemową (datę komputera).

Przykład:

```
$dataDok = DATE( [data dokumentu] )
```

MONEY(wartość)

Polecenie dokonuje zamiany kwoty z formatu pliku wejściowego (określonego poleceniem #SEPARATOR_01), do postaci liczby bez separatorów tysięcy i z kropką jako separatorem dziesiętnym.

Przykład:

```
$kwota = MONEY( [kwota dokumentu] )
```

SWITCH(wyrażenie)

```
{  
  CASE wartość  
    Blok poleceń  
  CASE wartość  
    Blok poleceń  
  [CASE ELSE]  
    Blok poleceń  
}
```

Polecenie powoduje obliczenie wyrażenia oraz wykonanie bloku poleceń skojarzonych z wynikiem obliczenia.

Przykład:

```
SWITCH( [stawka vat] )
{
CASE „22”
    $vat1 = [vat]
    BREAK
CASE „7”
    $vat2 = [vat]
    BREAK
CASE ELSE
    ERROR("Nierozpoznana stawka VAT: ", [stawka vat])
}
```

BREAK

Polecenie powoduje przerwanie wykonania bloku poleceń WHILE oraz SWITCH

POS([pole])

Polecenie bez parametru zwraca aktualną pozycję w pliku wejściowej (lina, pozycja w linii) bieżącej sekcji wejściowej.

Polecenie z podanym parametrem zwraca aktualną pozycję w pliku wejściowym (lina, pozycja w linii) danej o podanej nazwie. Pozycja jest podawana w postaci tekstu: „linia, pozycja”

Przykład:

```
ERR("Niekompletne dane sekcji: ", NAME(), " pozycja: ", POS() )
```

SECTION(nazwa sekcji we, nazwa sekcji wy [, opcje sekcji])

Jest to polecenie blokowe, grupujące zbiór poleceń przetwarzających jedną, określoną sekcję wejściową. Wewnątrz sekcji mogą znajdować się kolejne sekcje (zwane podsekcjami), grupujące polecenia przetwarzające odpowiednie podsekcje sekcji wejściowej. Polecenie tworzy opcjonalnie sekcję (podsekcję) wyjściową.

Nazwa sekcji we - służy do połączenia właściwej sekcji szablonu z właściwą sekcją wejściową. Sekcja wejściowa o określonej nazwie jest przetwarzana sekcją szablonu o identycznej nazwie. Jeśli w szablonie nie ma sekcji o zgodnej nazwie, wówczas wykonywana jest sekcja o nazwie pustej. Jeśli i takiej sekcji nie ma, wówczas zgłaszany jest błąd wykonania.

Nazwa sekcji wy - jest to nazwa podsekcji wyjściowej, która jest tworzona w wyniku wykonania kodu sekcji. Jeśli argument ten jest pusty, wówczas sekcja wyjściowa nie jest tworzona.

Opcje sekcji - są istotne tylko dla podsekcji, opisują dodatkowe akcje związane z wykonaniem sekcji:

FIND – powoduje odszukanie w danych wejściowych podsekcji o podanej nazwie.

REPEAT – otwierane i wczytywane są wszystkie kolejne podsekcje wejściowe o podanej nazwie

Argumenty te mogą być łączone.

Przykłady sekcji znajdują się na końcu opracowania.

MAKESEC(nazwa sekcji wyjściowej)

Polecenie tworzy nową sekcję (podsekcję) wyjściową o podanej nazwie. Jest to polecenie blokowe, grupujące polecenia odnoszące się do nowo utworzonej sekcji wyjściowej. Po wykonaniu bloku poleceń sekcja wyjściowa jest zamykana i od tego momentu jest już niedostępna.

Przykład:

```
IF( EQUAL( NAME(), „Dostawca” ) )
    MAKESEC( „Kontrahent” )
{
    $nazwa = [nazwa kontrahenta]
}
```

EXIT()

Polecenie powoduje opuszczenie (zamknięcie) bieżącej sekcji wejściowej, bez zamykania sekcji wyjściowej. Umożliwia to m.in. połączenie danych kilku sekcji wejściowych w dane jednej sekcji wyjściowej. Polecenie Exit odnosi się tylko do jednego poziomu zagnieżdżenia podsekcji, co oznacza że powoduje wyjście tylko z jednego poziomu zagnieżdżenia, a więc użyte w podsekcji wejściowej spowoduje wyjście z tej podsekcji i pozostanie w sekcji głównej.

DEF(nazwa definicji [, wartość definicji])

Polecenie z jednym argumentem umożliwia pobranie aktualnej wartości podanej definicji globalnej. W przypadku braku podanej definicji zwracany jest pusty tekst.

Polecenie dwuargumentowe umożliwia zmianę istniejącej lub utworzenie nowej definicji, np. formatu daty lub kwoty, w trakcie wykonywania importu. Polecenie zwraca poprzednią wartość definicji.

Nazwa definicji – nazwa zmienianej definicji globalnej, np.:

DATA - odpowiada poleceniu globalnemu #DATA

SEPARATOR_01 - odpowiada poleceniu globalnemu #SEPARATOR_01

Wartość definicji – nowa wartość definicji.

UWAGA: Polecenie nie może być użyte do zmiany formatu pliku wejściowego.

Przykład:

```
DEF( "DATA", [format daty] )
```

Własne definicje mogą być wykorzystane jak o zmienne globalne, które nie są niszczone po zakończeniu przetwarzania jednej sekcji wejściowej, a więc zawierać ustawienia sterujące całym procesem importu specjalnego. Od wersji 4.0 programu Finanse i Księgowość wszystkie definicje można zmieniać w specjalnym oknie wywoływanym z formatki profili importu. Dzięki takiemu mechanizmowi można użyć definicji do określania np. numerów kont dla automatycznej dekretacji dokumentów. Prostota obsługi tego okna pozwala na zmianę tych kont przez księgowego, bez konieczności ingerowania edytorem tekstu bezpośrednio w zawartość plików szablonu.

RETURN(format_danych_zwrotnych, nazwa_pliku, rodza_danych)

Polecenie specyfikuje format pliku informacji zwrotnych odsyłanych do nadawcy danych, nazwę tego pliku oraz określa rodzaj tych informacji. Informacje zwrotne odsyłane są po zakończeniu importu.

Istnieją dwie zasadnicze grupy informacji zwrotnych: informacje o uzgodnieniach, oraz informacje o nie rozliczonych transakcjach. Pełny opis jest zamieszczony przy poleceniu globalnym #RETURN. W odróżnieniu od polecenia globalnego to polecenie może być umieszczone w sekcji szablonu (Zwykle w sekcji danych globalnych lub w sekcji <init>)

9 Wzorcowe szablony importu specjalnego

9.1 Szablony aktualnie wykorzystywane

9.1.1 FK2005.AMS

Szablon służy do wczytywania danych w formacie AMS, który został opracowany jako podstawowy format wymiany danych pomiędzy programami systemu Symfonia. Obsługuje wszystkie rodzaje informacji możliwych do zaimportowania do programu FK:

Dokumenty księgowe, w tym

Dekretacje

Rejestry VAT

Rozrachunki

Kartoteki programu FK

Kontrahenci

Pracownicy

Urzędy

9.1.2 SRTRW.AMS

Szablon służy do wczytywania danych w formacie AMS z programu Symfonia Środki Trwałe

9.1.3 PLACE.AMS

Szablon służy do wczytywania danych w formacie AMS z programu Symfonia Płace

9.1.4 HMF.K.AMS

Szablon służy do wczytywania danych w formacie AMS z programu Symfonia Handel Premium.

Poniżej obszernie omówienie poszczególnych instrukcji

Blok definicji

```
#opis "Dane z programu Handel, format FK 3.0", "*.TXT (Pliki tekstowe)" //
#format "AMS" //
#opisy "" //
#Disable "2" //poziom ostrzeżeń 1 (m.in. bez ostrzeżenia "Pole nie
znaleziono")
#POWIELONE_NAZWY "0" //0-zapisuje bez pytania; 1-pyta czy zapisać; 2-nie
zapisuje
```

Tworzenie sekcji wyjściowej kontrahenta (w FK), gdy napotkamy sekcję wejściową (w pliku danych) o tej samej nazwie. Sekcja zostanie utworzona niezależnie od tego w jakim poziomie zagłębienia sekcji wejściowych będzie się znajdowała.

```
Section( "Kontrahent", "Kontrahent")
{
    $klucz = SUM0([id],-65535)
```

Nadawca jako klucza powiązania używa id ze swojego pliku danych zmniejszonego o 65535 (o offset do 1-ego numeru)

```
$skrot = [kod]
$nazwa = [nazwa]
$miestowosc = [miestowosc]
$gmina = [rejon]

IF ( REX( [nip], "{[rRpP]}/{*}" ) )
{
    IF ( EQUAL( TRANSLATE(@rex1, "U"), "P") )
        $pesel = @rex2
    ELSE
        $regon = @rex2
}
ELSE
{
    $nip = [nip]
}
```

W powyższym ciągu instrukcji na podstawie zawartości pola nip w pliku danych nadawca decyduje czy w FK odpowiada to nipowi, peselowi czy też regonowi

```
IF( EXIST( "dom" ) )
{
    $ulica = [ulica]
    $NumerDomu = [dom]
    $NumerMieszk= [lokal]
}
ELSE
{
    IF( REX([ulica], "-{*}([0-9///-]##:b()|[A-Q])")(:b[/\\.\,](m[,.\+)]() :b{[0-9]##:b[A-Z]()}|()$"))
    {
```

```

        $ulica      = @rex1
        $NumerDomu  = @rex2
        $NumerMieszk= @rex3
    }
    ELSE
        $ulica      = [ulica]
    }

    $kod            = [kodpocz]
    $telefon1       = [tel1]
    $telefon2       = [tel2]
    $telefax        = [fax]
    IF( equal([badres], "") )
        $w_banku1   = [bnazwa]
    ELSE
        $w_banku1   = CAT([bnazwa], ", ", [badres])
    $rachunek1      = [bkonto]
    $nazwisko       = [nazwisko]
    $imie           = [imie]
    $fk_ident       = [khfk]

    $regon         = [regon]
    $pesel          = [pesel]

    $idKraj         = [idKraj]
    $krajKod        = [krajKod]

    $statusUE       = [statusUE]

    $limit          = [limit]
    $limitKwota     = [limitKwota]
    $limitWaluta    = [limitWaluta]
    $znacznik       = [znacznik]

    $www            = [www]
    $email          = [email]

    $aktywny       = [aktywny]
}

Section( "Pracownik", "Pracownik")
{
    $klucz          = SUM0([id], -65535)
    $nazwisko       = [nazwisko]
    $imie1          = [imie]
    $imie2          = [naglowek] //imie2
    $telefon        = [tel1]
    $st_miejscowosc = [miejscowosc]

    $st_kod         = [kodpocz]

```

```

$st_poczta      = [miejscowosc]
$rch_symbol     = [bnazwa]
$rch_numer      = [bkonto]

$aktywny       = [aktywny]
$skrot          = [kod]
}

Section( "Urząd", "Urząd")
{
    $skrot        = [kod]
    $klucz        = SUM0([id],-65535)
    $nazwa        = [nazwa]

    IF( EXIST( "dom") )
    {
        $ulica     = [ulica]
        $NumerDomu  = CAT([dom], " / ", [lokal])
//    $NumerMieszk= [lokal]
    }
    ELSE
    {
        IF( REX([ulica], "-{*}([0-9///-]##:b()|[A-
Q]))(:b[/\\.\,]|(m[,.]+)|( ) :b{[0-9]##:b[A-Z]|( )})|( )$"))
        {
            $ulica    = @rex1
            $NumerDomu = CAT(@rex2, " / ", @rex3)
//    $NumerMieszk= @rex3
        }
        ELSE
            $ulica    = [ulica]
    }

    $kod          = [kodpocz]
    $miejsce      = [miejscowosc]
    $symbol       = [bnazwa]
    $rachunek     = [bkonto]

    $aktywny      = [aktywny]
}

Section("dokument", "")
{
    if( NOT(equal( [anulowany], "1")) )
    {
        MakeSec("dokument")
        {
            @opisy = ""
            Info("Dokument: ", [kod])
            $ObslugujJak = [obsluguj jak]
        }
    }
}

```

```

$klucz    = [symbol FK]
$nazwa    = [FK nazwa]
$rodzaj_dok  = [rodzaj_dok]
$naliczenie_VAT = [naliczenie_VAT]

switch( [obsluguj jak] )
{
case "RK"
    Info("konto kasy: ", [KontoKasy])
    if( equal([KontoKasy], "") )
    {
        Ask("Dokument nie posiada zdefiniowanego konta kasy.",
            "Ponieważ nie można określić strony dla konta kasy,",
            "pole kwoty pozostanie puste. DefBtn:&OK=2 Ico:!")
    }
case "DP"
case "WB"
case "DIM"
case "DEX"
    @opisy = 1
    $tresc = [opis FK]
    break
case else
    if (!Def("opisy"))
    {
        Def("opisy", Ask("Czy importować opisy faktur VAT i rachunków?
DefBtn:&Tak=10 Btn:&Nie=C Ico:?" )
    }
    if( Equal( Def("opisy"), "10") )
    {
        @opisy = 1
        $tresc = [opis FK]
    }
}

if ( Equal([rodzaj_dok], "zakupu") )
{
    if ( Equal([obsluguj jak], "WNT") )
    {
        $DataDok = Date([data])
        $DataOper = Date([data])
    }
    else
    {
        $DataDok = Date([datasp])
        $DataOper = Date([datasp])
    }
}
else
{

```

```

    if ( Equal([rodzaj_dok], "magazynowy") )
    {
        $DataDok = Date([datasp])
        $DataOper = Date([data])
    }
    else
    {
        $DataDok = Date([data])
        $DataOper = Date([datasp])
    }
}

$Kwota    = [kwota]

$Sygnatura = [Sygnatura]

$SaldoPoczRK = [SaldoPRK]
// $SaldoZapRK = sum0([SaldoZRK], [SaldoPRK])
$SaldoZapRK = [SaldoZRK]
$KontoKasy = [KontoKasy]

$NazwaKor = [NazwaKor]
$DataKor = Date([DataKor])
$Datawpl = Date([Datawpl])
IF(NOT (equal([plattermin], "")) )
{
    @plattermin = Date([plattermin])
}

switch( [obsluguj jak] )
{
case "DIM"
    break
case else
    IF ( NOT(equal([waluta], "")) )
    {
        if ( !equal([wkwota], "")) )
        {
            $wkwota = [wkwota]
            $waluta = [waluta]
            $kurs = [kurs]
            $typkursu = 3
        }
    }
}

//analizujemy wszystkie sekcje - mogą być wymieszane
Section("", "", "REPEAT") //REPEAT: analizujemy wszystkie podsekcje
{
    switch( Translate( Name(), "L") )

```

```

{
case "dane nabywcy"
case "dane sprzedawcy"
  if (!equal([khid], 0))
  {
    MakeSec("Kontrahent")
    {
      $klucz = SUM0([khid],-65535)
      if (!Equal([fk_ident], "0"))
        $fk_ident = [fk_ident] //uzgodniony numer FK
      IF ( REX( [khnip], "{[rRpP]}/*$" ) )
      {
        // na dokumencie piszemy tylko NIP , więc nic nie robimy
      }
      ELSE
      {
        $NIP = [khnip]
      }
    }
  }
else
{
  if( !equal( [khnip], "" ) )
  {
    MakeSec("Incydentalny")
    {
      $klucz = [khnip]
      $NIP = [khnip]
      $nazwa = [khnazwa]
      $miejscowosc= [khmiasto]
      $ulica = [khadres]
      IF( EXIST( "khdom" ) )
      {
        $NumerDomu = [khdom]
        $NumerMieszk= [khlokal]
      }
      $kod = [khkodpocz]
    }
  }
  else
  if( !equal( [khnazwa], "" ) )
  {
    MakeSec("Incydentalny")
    {
      $klucz = [khnazwa]
      $NIP = [khnip]
      $nazwa = [khnazwa]
      $miejscowosc= [khmiasto]
    }
  }
}

```

```

        $ulica      = [khadres]
        IF( EXIST( "khdom" ) )
        {
            $NumerDomu = [khdom]
            $NumerMieszk= [khlokal]
        }
        $kod          = [khkodpocz]
    }
}
break
case "zapis"
    MakeSec("zapis")
    {
        $strona      = Rex([strona], "MA", 1) //WN=0 (""), MA=1
        $kwota        = [kwota]
        $Konto        = [Konto]
        $NumerDok      = [NumerDok]
        $IdDlaRozliczen = [IdDlaRozliczen]
        if ( Equal(@@opisy, 1) )
        {
            $Opis      = [opis]
            $typopisu = 1 //opis bezpośrednio
        }
        $ZapisRownolegly = [ZapisRownolegly]
        $pozycja         = [Pozycja]

        IF ( NOT(equal([waluta], "")) )
        {
            $wkwota = [wkwota]
            $waluta = [waluta]
            $kurs = [kurs]
            $typkursu = 3
        }
    }
break
case "rejestr"
    IF( Equal(Equal($$obslugujJak, "FWN"), Equal($$obslugujJak, "DP")) )
    {
        MakeSec("rejestr")
        {
            IF( EXIST( "skrot" ) )
                $Klucz = CAT([skrot], ",", "ZW", ",", [stawka1])
            ELSE
                $Klucz = CAT($$klucz, ",", "ZW", ",", [stawka1])

            switch ( [ABC] )
            {
                case "2"
                    $ABC = 2 //B
            }
        }
    }

```

```

        break
    case "4"
        $ABC    = 4  //C
        break
    case "8"
        $ABC    = 8  //D
        break
    case else
        $ABC    = 1  //A
        break
    }

switch( Translate( $$rodzaj_dok, "L") )
{
    case "sprzedaży"
        switch( [metoda_vat] )
        {
            case "0"
                $oczek = 0
                break
            case "1"
                switch( $$ObslugujJak )
                {
                    case "FKS"
                    case "RKS"
                        $oczek = 0
                        break
                    case else
                        $oczek = 2
                        break
                }
                break
            case "2"
                $oczek = 0
                break
        }
        break
    case "zakupu"
        switch( $$naliczenie_VAT )
        {
            case "0"
                $oczek = 0
                break
            case "1"
                $oczek = 1
                break
            case "3"
                $oczek = 2
                break
            case "4"

```

```

        $oczek = 2
        break
    case "5"
        $oczek = 2
    }
    break
}

$okres    = [okres]
$netto    = [netto1]
$vat      = [vat1]
$brutto   = [brutto1]
$stawka  = [stawka1]
$sumanetto = [sumanetto]
$sumavat  = [sumavat]
// $brutto = SUM2([sumanetto], [sumavat])
// $podatek = [37] //Podatek;
// $dozaplaty = [38] //DoZapłaty;

$typ      = "ZW"
// ponizsze poznum od v5.20 mozna wyrzucic - robi to import
switch ( $$obslugujJak )
{
    case "FVS"
    case "FVZ"
    case "RUS"
    case "RUZ"
        $poznum = -1
        break
}
}
}
break
case "transakcja"
    MakeSec("transakcja")
    {
        $ustawowe    = 1 //odsetki ustawowe
        $zaliczka     = 0 //Zaliczka
        $idDlaRozliczen = [idDlaRozliczen]
        $termin       = [termin] //Termin;
    }
    break
case "rozliczenie"
    MakeSec("rozliczenie")
    {
        $dSymbol = [dSymbol] //numer własny dokumentu
        $idDlaRozliczen = [idDlaRozliczen]
        if( !equal([kwota], "") )
        {
            $kwota = [kwota]
        }
    }
}

```



```

Section("Z oddziału. Dok. pieniężny", "")
{
}

Section("Waluta", "")
{
}

```

Na koniec bardzo ciekawa instrukcja informująca, że jeśli pojawi się nowa sekcja danych, to proces importu powinien podnieść czujność importującego dane. Ma to miejsce również w przypadku gdyby doszło do przekłamania nazwy sekcji.

```

Section("", "")
{
    Warning("Nierozpoznana sekcja: ", Name())
}

```

9.1.5 HMFFK.AMS

Szablon służy do wczytywania danych z programu Handel ERP (format AMS).

9.1.6 EFAKTURY.AMS

Szablon służy do wczytywania danych z programu Symfonia e-Faktury EDS. Plik wejściowy dla szablonu tworzony jest za pomocą szablonu eFaktury.xslt

9.1.7 Kiedy definiować własne szablony importu?

Zawsze. Zasada odmienna niż przy formacie danych, które na ogół nie ewoluują. Szablony dostarczane przez producenta dotyczą specyficznej postaci danych często związanych z konkretną wersją. W każdej chwili, z kolejnym patchem postać przetwarzania danych może się zmienić. Dlatego też niektóre z tych par : szablon + pliki wymiany obsługują we własny sposób wersjonowanie. Porównaj daną [Wersja szablonu] np. w hmffk.ams. Przy wykorzystaniu szablonu producenta do obsługi własnych plików danych narażamy się na to że w kolejnej wersji patcha import może przestać działać prawidłowo. Jak zatem należy postępować?

W przypadku tworzenia nowego importu z obcej aplikacji, wydaje się że najbezpieczniej jest dopasować postać eksportowanych danych do postaci tożsamej z jakimiś danymi obsługiwanymi przez Sage, a następnie skopiować szablon obsługujący ten plik na szablon własny o nowej nazwie. I w tym własnym szablonie obsługiwać już na swoje potrzeby wersjonowanie. Wówczas żadna zmiana szablonu producenta nie zaburzy nam naszego importu, a możemy w nim zgodnie z wymogami zarządzać zmianą wersji, obsługą własnych definicji, etc.

9.1.8 Kiedy definiować własne szablony XSLT?

Powinno się unikać definiowania własnych szablonów XSLT na obecnym etapie. Szablon umożliwia wykorzystywanie zewnętrznych źródeł danych w formacie XML. Prawidłowo przygotowany schemat XSLT powinien umieć wygenerować z wejściowego pliku XML odpowiedni plik TXT, który następnie powinien zostać importowany za pomocą nowo-zdefiniowanego szablonu AMS.

9.2 Szablony historyczne

W poniższej części opisane są przykładowe szablony, które umożliwiały importowanie danych z dawnych, aktualnie już nieużywanych wersji różnych programów rynkowych. Traktować je należy wyłącznie jako przykłady instrukcji wczytywania różnych niestandardowych danych.

Uwaga! Nie będą one działały ze współczesnymi odpowiednikami tych programów ze względu na znaczne różnice w eksportowanych plikach.

9.2.1 SYM200.AMS

Szablon SYM200.AMS służy do importu danych w formacie AM i najlepiej był „rozumiany” przez wersje FK3.0 i FK 4.0. Wraz z przechowaniem coraz bardziej złożonych informacji w strukturach danych szablon ten nie jest w stanie sprostać bardziej złożonym dokumentom. Jest używany przez program Symfonia Sprzedaż, oraz mógł być używany w początkowych wersjach programu Handel oraz Środki Trwałe. Jest to szablon złożony, uruchamiający dodatkowe szablony dla wczytywania danych z wielu plików tworzących komplet danych.

Właściwe procedury importu danych realizowane są przez następujące szablony pomocnicze:

Dokum.amx - wczytanie dokumentów
Kontrah.amx - wczytanie kontrahentów
Pracow.amx - wczytanie pracowników
Rachba.amx - wczytanie rachunków bankowych
Urząd.amx - wczytanie urzędów skarbowych

Ponieważ jest to modyfikacja popularnego formatu SDF często był używany jako format do którego przystosowują się inne programy spoza systemu Symfonia.

Pliki danych

Pliki używane przy imporcie danych w formacie AM są następujące:

Setup.ini

Zawiera informacje o nadawcy danych, wszystkie istotne informacje z tego pliku mogą zostać podane bezpośrednio w profilu importu specjalnego. Obecnie plik ten służy głównie do identyfikacji obecności nowych danych.

e_dokum.txt

Zawiera informacje o dokumentach, zapisach, rejestrach VAT, transakcjach i rozliczeniach. Typ informacji wyróżniany jest przez pierwsze pole w rekordzie (np. **DOK, VAT, ZAP**).

e_kontr.txt

Zawiera informacje o nie uzgodnionych kontrahentach. Plik może nie występować, generowana jest wówczas informacja o braku pliku.

e_pracow.txt

Zawiera informacje o nie uzgodnionych pracownikach. Plik może nie występować, generowana jest wówczas informacja o braku pliku.

e_rachba.txt

Zawiera informacje o nie uzgodnionych rachunkach bankowych. Kolejność i znaczenie pól jest takie samo jak przy tekstowym eksporcie tabeli rachunków w programie FK. Plik może nie występować, generowana jest wówczas informacja o braku pliku.

e_urzad.txt

Zawiera informacje o nie uzgodnionych urzędach. Plik może nie występować, generowana jest wówczas informacja o braku pliku.

Pliki zwrotne

Po dokonaniu importu mogą zostać wygenerowane pliki zawierające informacje zwrotne do programu

i_rozlic.txt nie został nigdy wykorzystany i jest zawsze pusty

i_kartot.txt informacje o dokonanych uzgodnieniach pozycji kartotek

9.2.2 POLKA.AMS

Jest to szablon przystosowany do importu danych z programu POLKA. Obsługuje format importu specjalnego stosowanego w programach Finanse i Księgowość w wersji do 2.0 włącznie. Specyfikację pól danych w tym formacie można uzyskać od producenta programu POLKA.

9.2.3 CHART.AMS

Jest to szablon przystosowany do wczytywania danych z programu CHART firmy MIKROTEL. Jest to program fakturujący współpracujący z elektronicznymi centralami telefonicznymi tej samej firmy. Program dedykowany jest m.in. do obsługi hoteli.

9.2.4 SUBIEKT.AMS

Jest to szablon przystosowany do współpracy z programem SUBIEKT firmy INSERT. Szczegóły dotyczące formatu tego pliku można uzyskać w firmie INSERT.

9.2.5 CDN_FPP.AMS

Jest to szablon przystosowany do importu danych z programu Firma++, którego producentem jest firma CDN. Szczegóły dotyczące formatu danych można uzyskać w firmie CDN. Ten szablon umożliwia wczytywanie faktur oraz rachunków uproszczonych

9.2.6 CDN_FPPK.AMS

Jest to szablon przystosowany do importu danych z programu Firma++, którego producentem jest firma CDN. Szczegóły dotyczące formatu danych można uzyskać w firmie CDN. Ten szablon umożliwia wczytywanie raportów kasowych zawierających wszystkie KP/KW z jednego dnia.

10 Specyfikacja sekcji wyjściowych dla programu Finanse i Księgowość

10.1 Specyfikacja pól dla sekcji „Dokument”

Typ pola:

- W** Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
A wypełniane automatycznie przez import, zalecane nie wypełnianie
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	Opis pola
W	Klucz	Unikalny identyfikator rodzaju dokumentu
Z	Id	Unikalny identyfikator rekordu
Z	Errors	Bitowe znaczniki błędów w dokumencie
	Skrot[4]	Skrót numeru ewidencyjnego dokumentu. Musi być to predefiniowany skrót dokumentu wzorcowego lub zdefiniowanego przez użytkownika
A	numer	Kolejny numer ewidencyjny w ramach podanego skrótu.
W	Nazwa[40]	Nazwa własna dokumentu
	Tresc[60]	Treść dokumentu
Z	Datawpr	Data wprowadzenia / ostatniej modyfikacji
W	Datadok	Data dokumentu
W	Dataokr	Data okresu sprawozdawczego
W	DataOper	Data operacji gospodarczej, o ile występuje na formatce danego dokumentu
W	Datawpl	Data wpływu dla dokumentów zakupu, o ile występuje na formatce danego dokumentu
Z	DataRez	nie używane
	Kwota	Kwota dokumentu bez księgowani równoległych
	Wkwota	kwota w walucie dla dokumentów jednowalutowych
	KwotaZRow	kwota całkowita dokumentu z księgowaniami równoległymi
1)	Tabela	Identyfikator rekordu tabeli kursów walut dla dokumentu jednowalutowego
1)	Kurs	Kurs waluty z dokładnością do 6 miejsc po przecinku
1)	Typkursu	Rodzaj kursu waluty dla dokumentu jednowalutowego: 0-bez waluty 1-zakup 2-sprzedaż 3-średni
1)	Waluta[4]	Nazwa waluty dla dokumentu jednowalutowego
	NazwaKor[40]	Nazwa własna dokumentu korygowanego
	DataKor	Data dok. Korygowanego
Z	Rezerwa1	nie używane, musi być zero
Z	Sygnatura	Sygnatura osoby dokonującej modyfikacji w buforze lub osoby księgującej w księgach
Z	Kontrahent	identyfikator rekordu kontrahenta w bazie kontrahentów stałych lub incydentalnych. Wypełniane na podstawie sekcji „Kontrahent”
Z	Rezerwa2	nie używane, musi być zero

Z	nip[20]	Numer NIP kontrahenta stałego lub incydentalnego, Wypełniane na podstawie sekcji „Kontrahent”
Z	Automat	znacznik dokumentu księgowanego automatycznie (np. różnice kursowe), w przeważającej części nie powinno być wypełniane
Z	Rezerwa3	nie używane, musi być zero
Z	OkresDK	Numer okresu dziennika księgowania, tylko dla dokumentów zaksięgowanych. Dla bufora musi być zero
Z	NumerDK	Numer kolejny w dzienniku księgowania, tylko dla dokumentów zaksięgowanych. Dla bufora musi być zero
2)	SaldoPoczRK	Saldo początkowe raportu kasowego (+Wn, -Ma). Jeśli jako wartość tego pola wpisujemy tekst „oblicz”, wówczas program obliczy automatycznie saldo początkowe tego dokumentu jako równe saldu końcowemu ostatniego dokumentu tego typu w księgach lub buforze – jest to zalecana wartość tego pola.
2)	SaldoZapRK	Saldo zapisów raportu kasowego (+Wn, -Ma) Jeśli jako wartość tego pola wpisujemy tekst „oblicz”, wówczas program obliczy automatycznie saldo zapisów raportu kasowego – jest to zalecana wartość tego pola.
Z	Rezerwa4	nie używane, musi być zero. Nie występuje od wersji 4.0
Z	Rezerwa5	nie używane, musi być zero. Nie występuje od wersji 4.0
3)	KontoKasy	Numer konta kasy dla raportu kasowego
4)	KwotaPozaRej	Kwota dokumentu, która nie wchodzi do rejestru VAT. Wypełniane tylko dla rachunków i faktur VAT
4)	KwotaEuro	Kwota dokumentu w walucie EURO
4)	KursEuro	Kurs waluty EURO dla tego dokumentu
W 5)	ObslugujJak	Skrót numeru ewidencyjnego dokumentu wzorcowego (DP, FVZ, FVS, itp)

1) Wypełnione wszystkie pola tylko dla dokumentu jednowalutowego. Jeśli dokument nie posiada odwołań do istniejącej tabeli kursów walut, wówczas pole *tabela* musi zawierać 0 (zero)

2) Pola muszą być wypełnione tylko dla raportu kasowego. Jeśli wartość pola będzie równa „oblicz”, wówczas program automatycznie obliczy wartości tych pól

3) Pole może być używane zamiennie z polem **KontoRap** w sekcji **Zapis**, służy również do wskazania zapisu na konto pracownika w dokumencie Rozliczenie Zaliczki.

4) Pola wprowadzone w wersji 4.0 programu Finanse i księgowość

5) Aby użytkownik podczas pierwszego importu danych wiedział jakiego charakteru jest dokument: faktura czy wyciąg bankowy, czy raport kasowy - musi otrzymać informację na ten temat. To pole temu właśnie służy. Na podstawie tej zawartości użytkownik może sam już podjąć decyzję czy należy zdefiniować w FK nowy typ dokumentu oparty na podanym typie wzorcowym czy użyć typ już istniejący. Pole to pozwala uniknąć omyłek jaką byłoby np. wczytanie raportu kasowego jako dokumentu prostego. Lista wzorcowych typów dokumentów w FK to: DP, FVZ, FKZ, FVS, FKS, DIM, DEX, RUZ, RKZ, RUS, RKS, RK, WB, RZL, FWN, WNT, WDT, PWN.

10.2 Specyfikacja pól dla sekcji „Zapis”

Typ pola:

W Pole wymagane – musi być wypełnione

Z Pole zarezerwowane, musi pozostać niewypełnione

1...n Patrz właściwy numer uwagi na końcu zestawienia pól

Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	Opis pola
Z	Id	Unikalny identyfikator rekordu zapisu
	IdDlaRozliczen	Dowolny, większy od zera, mniejszy od 30000, unikalny w ramach dokumentu identyfikator zapisu dla rozrachunków oraz rejestrów VAT raportu kasowego

Z	Dokument	Identyfikator dokumentu z którego pochodzi zapis
W	Pozycja	Numer kolejny bilansującej się grupy zapisów opatrzonych wspólnym opisem, numeracja jest ciągła, kolejna, rozpoczynająca się od 0.
Z	Rozbicie	Numer kolejny dekretu w ramach bilansującej się grupy zapisów (patrz <i>pozycja</i>). Numeracja dla każdej grupy jest ciągła, kolejna, rozpoczynająca się od 0. Numer 0 posiada pierwszy dekret po stronie WN, nr 1 - pierwszy dekret po stronie MA, 2,3...n kolejne dekrety (po WN lub MA). Pole pozostawione puste zostaje automatycznie wypełnione przez program.
Z	Typopisu	musi być równe 1
Z	typRozb	Określa sposób rozbicia kwoty po jednej stronie na częściowe dekrety: 0-nie ma rozbicia, 1-rozbicie po WN, 2-rozbicie po MA
W	Kwota	kwota zapisu w ZŁ
1)	Wkwota	kwota zapisu w walucie
W	zapisRownolegly	<p>Znacznik księgowania równoległego – podana wartość jest złożeniem bitowym poszczególnych atrybutów (niżej wartości atrybutów podane dziesiętnie)</p> <p>0 (wartość domyślna) – zapis nie jest równoległy, ani nie tworzy innych równoległych</p> <p>1 to jest zapis równoległy</p> <p>2 to jest zapis zwykły, dla którego utworzono zapis równoległy</p> <p>4 to jest zapis zwykły, który utworzył księgowanie równoległe automatyczne</p> <p>8 to jest zapis równoległy po stronie WN - dla RK</p> <p>16 to jest zapis równoległy po stronie MA - dla RK</p> <p>32 to jest zapis równoległy na życzenie</p> <p>64 równoległy zapis skojarzony z zapisem z pozycji bezpośrednio poprzedzającej zapis równoległy</p> <p>Sposób podania danych najlepiej oglądać w przykładzie do szablonów Symfonii, lub bezpośrednio w szablonach współpracy programów Symfonii dostarczanych z aplikacją.</p>
W	Strona	Strona dekretu: 0 - WN, 1 –MA
Z	Automat	zapis dokonany automatycznie, dla bufora nie używany
2)	kontoRap	<p>Tylko dla raportu kasowego: wartość niezerowa (1) oznacza zapis na konto kasy dla raportu kasowego.</p> <p>Wszystkie zapisy z ustawionym tym znacznikiem muszą mieć identyczny numer konta</p>
Z	Dataokr	data okresu sprawozdawczego, do którego zakwalifikowano dokument
Z	Datazap	nie używane
1)	waluta[4]	waluta dla zapisu walutowego (skrot ISO)
	opis[60]	opis pozycji
	Konto	<p>Numer konta. Dla poziomu analityki z kartoteki musi być podany prefix kartoteki oraz wartość klucza odpowiedniej pozycji z kartoteki, np. 201-K345</p> <p>Prefiksy:</p> <p>K – kontrahenci</p> <p>P – pracownicy</p> <p>R – rachunki bankowe</p> <p>U – urzędy</p> <p>Od wersji 2008 wartością klucza jest numer pozycji z danej kartoteki</p>
Z	Syntetyka	numer syntetyki konta

Z	poziom1	numer pierwszego poziomu analityki lub 0 gdy nie ma tego poziomu
Z	poziom2	numer drugiego poziomu analityki lub 0 gdy nie ma tego poziomu
Z	poziom3	numer trzeciego poziomu analityki lub 0 gdy nie ma tego poziomu
Z	poziom4	numer czwartego poziomu analityki lub 0 gdy nie ma tego poziomu
Z	poziom5	numer piątego poziomu analityki lub 0 gdy nie ma tego poziomu
1)	Tabela	identyfikator rekordu z tabeli kursów walut.
1)	Kurs	użyty kurs waluty
Z	Przeks	pole musi mieć wartość 0 dla zapisu złotówkowego oraz 2 dla walutowego
	numerDok[40]	numer dokumentu jako pozycji raportu kasowego
Z	NrRozbKP	określa sposób rozbicia konta przeciwstawnego dla bilansującej się grupy zapisów: dla zapisów po stronie WN: -1 oznacza rozbicie po stronie MA, zapis posiada wiele kont przeciwstawnych 1 oznacza, że rozbicie jest po stronie Wn, zapis posiada tylko jedno konto przeciwstawne 3 oznacza brak rozbicia, zapis posiada jedno konto przeciwstawne dla zapisów po stronie MA: -1 oznacza rozbicie po stronie WN, zapis posiada wiele kont przeciwstawnych 0 oznacza rozbicie po stronie MA, zapis posiada jedno konto przeciwstawne 2 oznacza brak rozbicia, zapis posiada jedno konto przeciwstawne
1)	Typkursu	rodzaj użytego kursu: 0- nie ma waluty 1-kurs zakupu 2-kurs sprzedaży 3-kurs średni
3)	KursEuro	Kurs waluty EURO, o ile zapis jest podany w EURO lub w walucie z koszyka, ważne dla transakcji obejmujących rozliczenia między walutą narodową a jej późniejszymi rozliczeniami w walucie €.
Z	Rezerwa1	nie używane
4)	uid	Tylko dla ERP. Przekazany globalny unikalny identyfikator (guid) jako string. Musi być unikalny w systemie.

1) Pola muszą być wypełnione dla zapisów (dokumentów) walutowych. Dla dokumentu jednowalutowego pola te dla wszystkich zapisów muszą być wypełnione identycznymi wartościami. Dla dokumentów wielowalutowych bilansująca się grupa zapisów (rozbicie) musi mieć identycznie wypełnione pola. Jeśli zapis nie posiada odwołania do istniejącej tabeli kursów, wówczas pole *tabela* musi mieć wartość 0 (zero)

2) Wyróżnienie konta kasy może odbywać się na dwa sposoby:

- przez ustawienie znacznika **KontoRap** w każdym zapisie na konto kasy. W tym przypadku wszystkie zapisy z niezerowym znacznikiem muszą mieć to samo konto.
- przez podanie w sekcji dokumentu numeru konta kasy w polu **kontoKasy**. W tym przypadku kontoKasy musi zgadzać się z każdym kontem podanym w zapisie wchodzącym na konto kasy.

3) Pole wprowadzone w wersji 4.0 programu Finanse i Księgowość

4) od wersji 2010.

10.3 Specyfikacja pól dla sekcji „Rejestr”

Typ pola:

W Pole wymagane – musi być wypełnione

Z Pole zarezerwowane, musi pozostać niewypełnione

1...n Patrz właściwy numer uwagi na końcu zestawienia pól

Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator dla definicji rejestru
Z	Id	unikalny identyfikator rekordu
Z	Def	identyfikator definicji rejestru
Z	dSkrot	skrót numeru ewidencyjnego dokumentu, dla którego jest rejestr
Z	dNumer	numer ewidencyjny dokumentu
Z	dId	identyfikator rekordu dokumentu
Z	rokDok	indeks roku, w który został zaksięgowany dokument
Z	okrDok	numer okresu sprawozdawczego dokumentu
Z	numerWl	numer własny dokumentu
Z	dataDok	data dokumentu
Z	dataWS	data wpływu dla zakupu, lub data sprzedaży
Z	Order	kolejność sortowania dla sprzedaży, określana na podstawie kolejnego numeru własnego
Z	Kontrahent	Numer pozycji kontrahenta: 1..999999 - kontrahent stały, 0 lub od 1000000 w górę kontrahent incydentalny
Z	Nip	nip, pesel lub regon kontrahenta
W	Abc	rodzaj zakupu dla rejestrów zakupu: A=1, B=2, C=4
W	Okres	data początku okresu rejestru (dzień = 1)
Z	vatOczek	znacznik, czy rekord przyszedł z rejestru VAT oczekującego
	dataKor	data dokumentu korygowanego
	numerKor(40)	numer własny dokumentu korygowanego
W	Brutto	kwota brutto
W	Netto	kwota netto
W	Vat	kwota VAT
W	Stawka	Stawka opodatkowania
	idRozliczenia	Musi być ustawione na 0
	Znacznik	Jak w dokumencie
	sOczek	
	Dozapłaty	kwota do zapłaty (brutto z zaokrągleniami itp)
Z	pozNum	numer łączący rejestr z zapisem w raporcie kasowym (równe wartości zapis. IdDlaRozliczen)

10.4 Specyfikacja pól dla sekcji „Transakcja”

Typ pola:

W Pole wymagane – musi być wypełnione

Z Pole zarezerwowane, musi pozostać niewypełnione

1...n Patrz właściwy numer uwagi na końcu zestawienia pól

Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	Opis / uwagi
1)	Stopa	Stopa odsetek stałych
1)	Ustawowe	Typ odsetek: 0-stała stopa, 1-odsetki ustawowe
	Zaliczka	0-nowa transakcja, 1-zaliczka
W	Termin	Termin płatności w postaci daty (terminu płatności) lub ilości dni licząc od daty dokumentu
2)	IdDlaRozliczen	Wartość idDlaRozliczeń z właściwego zapisu lub –1 dla rozrachunków zbiorczych dla faktur po właściwej stronie dla kontrahenta

1) Należy podać albo stopę odsetek stałych (dziennie) albo w pole ustawowe wstawić wartość 1
2) Pole musi być wypełnione tylko wówczas, gdy transakcja nie jest podsekcją zapisu. W obecnej wersji importu specjalnego dla jednego zapisu może być utworzona tylko jedna nowa transakcja albo jedno rozliczenie transakcji. Dodatkowe parametry, takie jak kwota, konto, waluta pobierane są z danych zapisu.

10.5 Specyfikacja pól dla sekcji „Rozliczenie”

Typ pola:

W Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	Opis / uwagi
W	DSymbol	Numer własny dokumentu rozpoczynającego transakcję
1)	IdDlaRozliczen	Wartość idDlaRozliczeń z właściwego zapisu lub –1 dla rozrachunków zbiorczych dla faktur
2)	kursWO	Kurs złotych względem waluty oryginalnej
2)	walutaObca	Skrót waluty oryginalnej rozliczanej transakcji
2)	rozliczonoWO	Kwota rozliczenia w walucie oryginalnej

1) Pole musi być wypełnione tylko wówczas, gdy rozliczenie nie jest podsekcją zapisu. W obecnej wersji importu specjalnego dla jednego zapisu może być utworzona tylko jedna nowa transakcja albo jedno rozliczenie transakcji. Dodatkowe parametry, takie jak kwota, konto, waluta pobierane są z danych zapisu.
2) Pola wypełniane jeśli prowadzimy rozliczanie między różnymi walutami. W komplecie tych pól należy podać wartości zgodne z walutą rozliczanej transakcji („oryginalnej”, ale obcej względem waluty zapisu)

10.6 Specyfikacja pól dla sekcji „Kontrahent”

Typ pola:

W Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator rekordu importowanego
Z	Id	Identyfikator rekordu
Z	Aktywny	
W	Skrot [40]	Unikalna skrócona nazwa
W	Nazwa [100]	Pełna nazwa
	Kraj [15]	Kraj kontrahenta zagranicznego
	Miejscowosc	Adres kontrahenta
	Gmina	j.w.

	Ulica	j.w.
	NumerDomu	j.w.
	NumerMieszk	j.w.
	Kod	j.w.
	Poczta	j.w.
	telefon1 [16]	Numer telefonu
	telefon2 [16]	Numer telefonu
	telefax [16]	Numer fax-u
	Telex [16]	Numer telex-u
Z	Rezerwa1	Nie używane
	W_banku1	Symbol rachunku bankowego banku
	Rachunek1	Numer rachunku bankowego
Z	Rezerwa2	Nie używane
	W_banku2	Symbol rachunku bankowego banku
	Rachunek2	Numer rachunku bankowego
Z	PlatnikVAT	Nie używane
Z	Naglowek	Nie używane
	Nazwisko	Osoba do kontaktów u kontrahenta
	Imie	Osoba do kontaktów u kontrahenta
Z	Pracownik	Nie używane
Z	Typ	Nie używane
	NIP	Numer NIP
	Pesel	Numer PESEL
	Regon	Regon
	Uwagi [100]	Uwagi

10.7 Specyfikacja pól dla sekcji „Incydentalny”

Typ pola:

- W** Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator rekordu importowanego
Z	Id	Identyfikator rekordu
W	Nazwa [100]	Pełna nazwa
	NIP	Numer NIP
	Miejscowosc	Adres kontrahenta
	Gmina	j.w.
	Ulica	j.w.
	NumerDomu	j.w.
	NumerMieszk	j.w.
	Kod	j.w.
	Poczta	j.w.

10.8 Specyfikacja pól dla sekcji „Pracownik”

Typ pola:

W Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola Oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator rekordu importowanego
Z	Id	Identyfikator rekordu
W	Nazwisko	
	Panienskie	
W	Imie1	
	Imie2	
	ImieOjca	
	ImieMatki	
	DataUrodz	
	MiejsceUrodz	
	Pesel	
	NIP	
	Telefon	Numer telefonu
	ST_Miejscowosc	Stały adres pracownika
	ST_Gmina	j.w.
	ST_Ulica	j.w.
	ST_NumerDomu	j.w.
	ST_NumerMieszk	j.w.
	ST_Kod	j.w.
	ST_Poczta	j.w.
	CZ_Miejscowosc	Czasowy adres pracownika
	CZ_Gmina	j.w.
	CZ_Ulica	j.w.
	CZ_NumerDomu	j.w.
	CZ_NumerMieszk	j.w.
	CZ_Kod	j.w.
	CZ_Poczta	j.w.
Z	UrządSkarbowy	Nie używane
	Rch_symbol	Symbol rachunku bankowego
	Rch_numer	Numer rachunku bankowego

10.9 Specyfikacja pól dla sekcji „Urząd”

Typ pola:

W Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola Oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator rekordu importowanego
Z	Id	Identyfikator rekordu

W	Nazwa	Nazwa urzędu
	Miejsce	Adres urzędu
	Ulica	j.w.
	NrDomu	j.w.
	Kod	j.w.
	symbol	Symbol rachunku bankowego
	rachunek	Numer rachunku bankowego

10.10 Specyfikacja pól dla sekcji „Rachunek”

Typ pola:

- W** Pole wymagane – musi być wypełnione
Z Pole zarezerwowane, musi pozostać niewypełnione
1...n Patrz właściwy numer uwagi na końcu zestawienia pól
Puste Pole może nie być wypełnione

Typ pola	Nazwa pola Oraz rozmiar pola	opis / uwagi
W	Klucz	Identyfikator rekordu importowanego
Z	Id	Identyfikator rekordu
W	Nazwa	Nazwa banku
	Kraj [15]	Kraj banku
	Miasto	Adres banku
	Ulica	j.w.
	telefon1 [16]	Numer telefonu
	telefon2 [16]	Numer telefonu
	telefax [16]	Numer fax-u
	Telex [16]	Numer telex-u
W	symbol	Symbol rachunku bankowego
W	numer	Numer rachunku bankowego
	Uwagi [100]	

11 Zasady wypełniania walut dla importu specjalnego.

11.1 Dokumenty bezwalutowe

Są to raporty kasowe oraz rozliczenia zaliczki pracownika. Te dokumenty **nie mogą** mieć wypełnionych informacji o walutach. Zalecane wartości odpowiednich pól są następujące:

```
Waluta    = //pusty tekst
WKwota    = 0
Tabela    = 0
Kurs      = 0
TypKursu  = 0
```

Te wartości powinny być ustawiane również dla pozostałych typów dokumentów, które nie mają wypełnionych walut.

11.2 Dokumenty jednowalutowe

Są to dokumenty proste, faktury oraz rachunki. Wszystkie pola dotyczące waluty i kursu (w sekcji dokument i we wszystkich sekcjach zapis) muszą być wypełnione **takimi samymi** wartościami, niezależnie czy poszczególne zapisy wchodzi na konta walutowe, czy też złotówkowe

Waluta = *symbol_waluty* //zgodnie z symbolem w tabeli kursów walut
WKwota = *kwotaPLN / kurs* //zaokrąglone do 1 gr

Tabela	= 0	//zawsze zero
Kurs	= <i>kurs</i>	//kurs waluty do 8 miejsc po przecinku
TypKursu	= <i>typKursu</i>	//1-zapup. 2–sprzedaż, 3–średni

11.3 Dokumenty wielowalutowe

Są to dokumenty importowe oraz wyciągi bankowe. Każda bilansująca się grupa zapisów może mieć swoje własne ustawienia dotyczące walut, jednak w ramach każdej grupy wszystkie pola dotyczące waluty i kursu muszą być wypełnione **takimi samymi** wartościami, niezależnie czy poszczególne zapisy wchodzą na konta walutowe, czy też złotówkowe

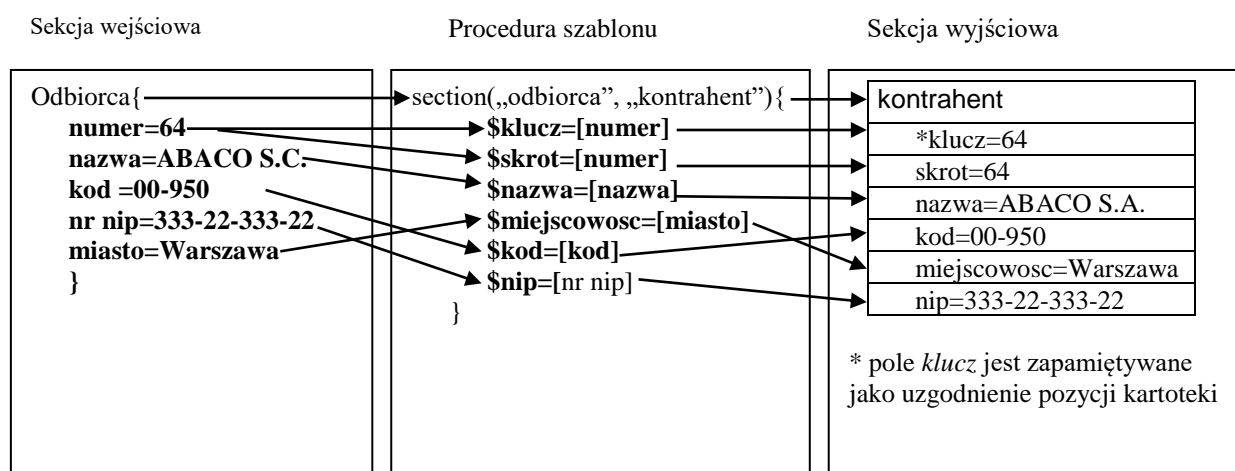
Waluta	= <i>symbol_waluty</i>	//zgodnie z symbolem w tabeli kursów walut
WKwota	= <i>kwotaPLN / kurs</i>	//zaokrąglone do 1 gr
Tabela	= 0	//zawsze zero
Kurs	= <i>kurs</i>	//kurs waluty do 8 miejsc po przecinku
TypKursu	= <i>typKursu</i>	//1-zapup. 2–sprzedaż, 3–średni

12 Przykłady szablonu i danych

12.1 Ilustracja procesu przetwarzania danych

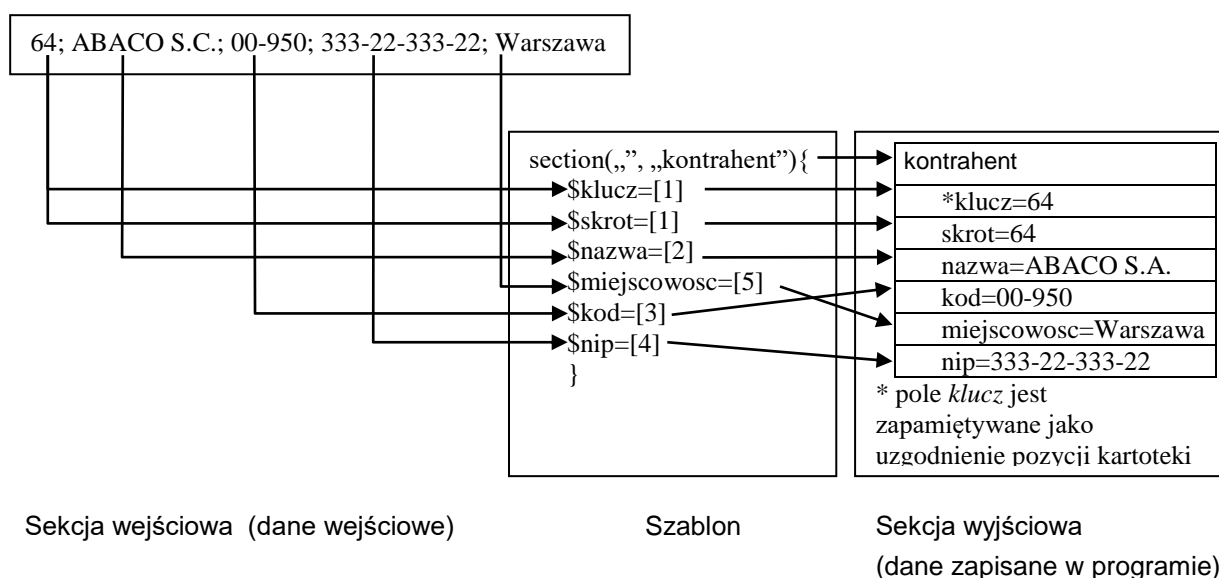
Przykłady ilustrują przebieg przetwarzania przez szablon danych wejściowych do postaci wymaganej przez procedury wewnętrzne programu. Strzałkami zaznaczono przykładowe przepływy danych.

W przypadku formatu z nazwami pól, kolejność pól w ramach jednej sekcji nie ma znaczenia. Również kolejność przetwarzania i analizowania pól w szablonie nie ma znaczenia. Wszystkie pola wejściowe są identyfikowane poprzez swoje nazwy.



Rysunek 4 Przetworzenie pojedynczej sekcji danych w formacie z nazwami pól

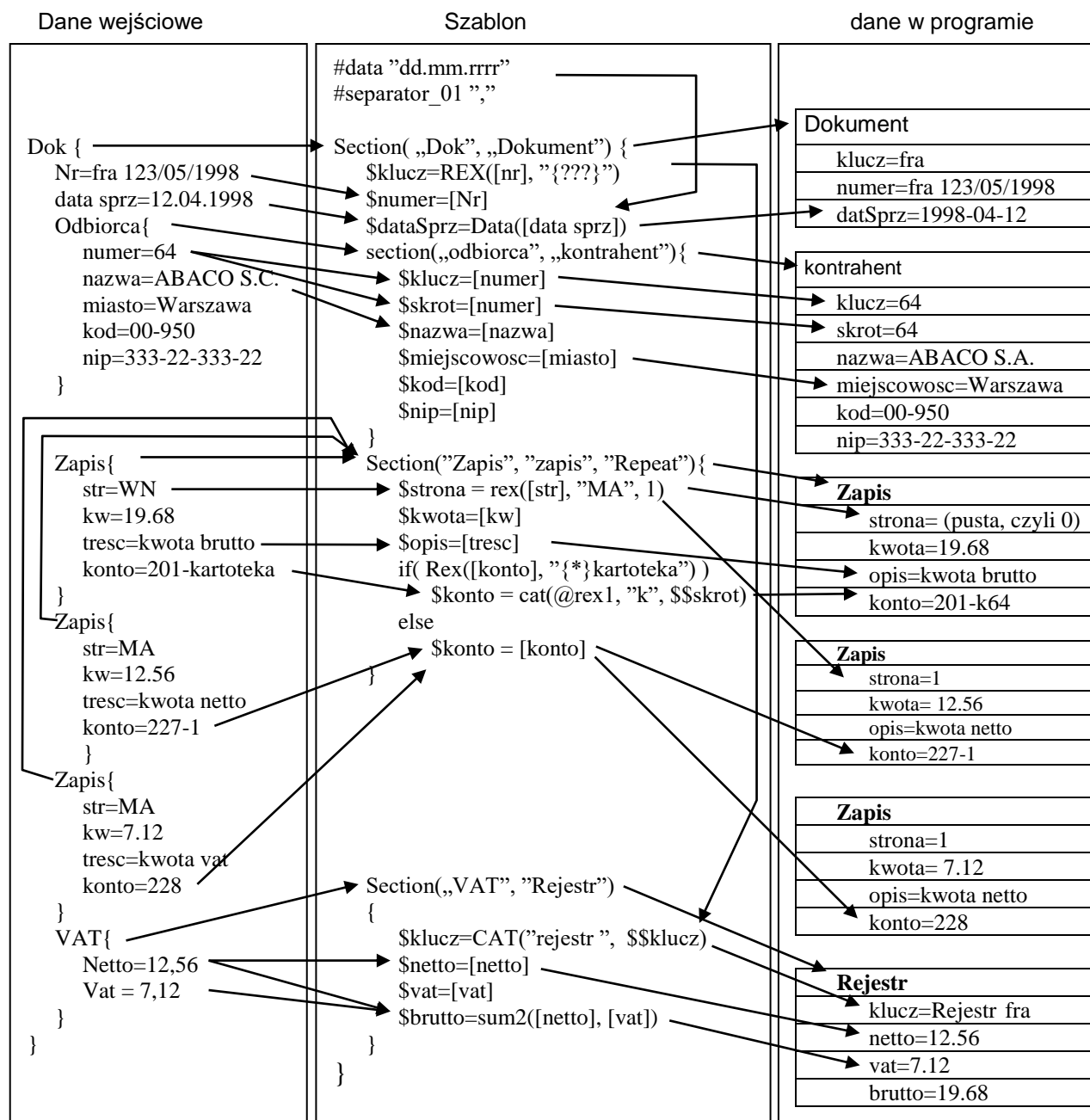
W przypadku formatu bez nazw pól (pola oddzielone separatorami), istotna jest kolejność pól w ramach jednej sekcji. Wszystkie pola identyfikowane są poprzez swój numer kolejny w linii danych. Kolejność przetwarzania i analizowania pól w szablonie nie ma znaczenia, do poszczególnych pól można odwoływać się wielokrotnie, w dowolnej kolejności. Dla formatu bez nazw pól zwykle jeden plik zawiera jeden rodzaj danych, a każda linia pliku jest jedną sekcją danych. W takim przypadku nie ma również jawnie podanej nazwy sekcji wejściowej. W szablonie podajemy wówczas pustą nazwę sekcji wejściowej.



Rysunek 5 Przetworzenie pojedynczej sekcji danych w formacie pól oddzielonych średnikami

W jednej sekcji danych może znajdować się wiele podsekcji o tej samej nazwie (np. wiele zapisów na różne konta w jednym dokumencie). W takim przypadku należy w szablonie zastosować pętlę programową w celu przeczytania wszystkich podsekcji. Można to zrobić dodając argument *REPEAT* do polecenia *SECTION*, lub

organizując klasyczną pętlę *WHILE*. Aby przetworzyć podsekcje o różnych nazwach (i zawierających różne dane), umieszczamy w szablonie odpowiednią liczbę podsekcji, przetwarzających odpowiadające im podsekcje wejściowe.



12.2 Proces przetwarzania EDS z wykorzystaniem XSLT

Profil importu EDS został oznaczony dodatkową flagą w pliku INI wymuszającą pobieranie plików wejściowych z aplikacji Symfonia e-Faktury EDS, zamiast standardowego przeszukiwania z katalogu. W przypadku utworzenia własnego profilu z wykorzystaniem własnego szablonu XSLT, pobieranie pliku następuje standardowo z katalogu.

Po etapie pobierania pliku wejściowego w formacie XML następuje przetworzenie za pomocą wskazanego w profilach szablonu XSLT (w przypadku profilu EDS jest to efaktury.xslt). Transformacja wejściowego XML zgodnie ze wskazanym szablonem XSL generuje tekstowy plik wyjściowy w katalogu tymczasowym. W następnym kroku tak wygenerowany plik TXT importowany jest zgodnie z ilustracją z pkt 12.1 na podstawie wskazanego w profilu szablonu AMS.

12.3 Przykład obsługi dokumentów walutowych

Poniższy przykład wykorzystuje format danych AMS

// fragmenty szablonu obsługującego dokumenty walutowe

```
Section("dokument", "dokument")
{
    .....
    $Kwota    = [kwota]
    //w sekcji „dokument” dodajemy wypełnianie pól dotyczących walut
    $Waluta= [waluta]
    $WKwota= [wkwota]
    $Tabela   = [tabela]
    $Kurs      = [kurs]
    $TypKursu = [typkursu]
    .....
    Section("", "", "REPEAT") //REPEAT: analizujemy wszystkie podsekcje
    {
        switch( Translate( Name(), "L") )
        {
            .....
            case "zapis"
                MakeSec("zapis")
                {
                    $kwota    = [kwota]
                    //w sekcji „Zapis” dodajemy wypełnianie pól dotyczących walut
                    $Waluta= [waluta]
                    $WKwota= [wkwota]
                    $Tabela   = [tabela]
                    $Kurs      = [kurs]
                    $TypKursu = [typkursu]
                }
                break
            case "transakcja"
                MakeSec("transakcja")
                {
                    //w transakcji wszystkie informacje są wypełniane automatycznie
                    $ustawowe = 1 //odsetki ustawowe
                    $zaliczka  = 0 //Zaliczka
                    $idDlaRozliczen = [idDlaRozliczen]
                    $termin    = [termin] //Termin;
                }
                break
            case "rozliczenie"
                MakeSec("rozliczenie")
                {
                    $dSymbol = [dSymbol] //numer własny dokumentu
                    $idDlaRozliczen = [idDlaRozliczen]
                    $kwota    = [kwota]
                    //w sekcji „rozliczenie” dodajemy kwotę walutową,
                    //kurs itd. pobierane są automatycznie z dokumentu lub zapisu
                    $w_kwota = [wkwota]
```

```

    }
    break
}
}
}

```

// fragment pliku danych zawierającego informacje walutowe

```

Dokument{
    ....
    kwota =1382.70
    wkwota =337.24
    waluta =USD
    kurs =4.1000
    tabela = 0
    typkursu=3
    Zapis{
        strona =WN
        kwota =1382.70
        wkwota =337.24
        waluta =USD
        kurs =4.1000
        tabela = 0
        typkursu=3
        konto =201-9
        IdDlaRozliczen =1
        ...
    }
    Zapis{
        strona =MA
        kwota =1133.36
        wkwota =276.43
        waluta =USD
        kurs =4.1000
        tabela = 0
        typkursu=3
        konto =731
        IdDlaRozliczen =2
    }
    ....
    }
    Zapis{
        strona =MA
        kwota =249.34
        wkwota =60.81
        waluta =USD
        kurs =4.1000
        tabela = 0
        typkursu=3
        konto =221-1
        IdDlaRozliczen =3
        ...
    }
}

```

```

}
Rozliczenie{
    IdDlaRozliczen= -1
    kwota =1382.70
    wkwota =337.24
    dSymbol=FV12/00/001
}
}

```

13 Kody błędów importu specjalnego

13.1 Kody błędów związane z szablonem

13.1.1 (1) Zbyt długi symbol: ...

Błąd występuje w przypadku użycia zbyt długiej nazwy zmiennej lub pola wyjściowego.

Maksymalna długość symboli w szablonie wynosi 60 znaków. Jeżeli błąd dotyczy symboli z pliku wejściowego, jedyną możliwością jest zmodyfikowanie tego pliku.

13.1.2 (2) Zbyt długie wyrażenie stringowe.

Błąd występuje w przypadku użycia zbyt długiej wartości tekstowej, podstawianej bezpośrednio do zmiennej, pola wyjściowego lub jako argument wbudowanej funkcji.

Maksymalna długość tekstów podanych bezpośrednio w szablonie wynosi 300 znaków. W przypadku łączenia tekstów poleceniem CAT maksymalna długość wynikowego tekstu nie może przekroczyć 5000 znaków

13.1.3 (3) Brak zamykającego cudzysłowu.

Podano argument tekstowy (podstawienia lub argumentu funkcji) bez zamykającego cudzysłowu.

13.1.4 (4) Pusta lub zbyt długa nazwa pola.

Błąd występuje tylko dla pól wejściowych i oznacza, że w nawiasach [] oznaczających pole wejściowe nie podano żadnych znaków różnych od spacji, pominięto nawias zamykający] lub podana nazwa pola przekracza dozwoloną długość 60 znaków.

13.1.5 (5) Nieznane polecenie: ...

W szablonie użyto nieznanego polecenia. Przyczyną może być błąd literowy w nazwie pola lub opuszczenie części nazwy (np. SUM zamiast SUM0). Błąd wystąpi wówczas, jeśli po symbolu występuje nawias otwierający, sugerujący wywołanie funkcji.

13.1.6 (6) Oczekiwany znak początku bloku.

Użyto polecenia wymagającego znaków { } początku i końca bloku (np. Switch) , ale po poleceniu nie podano znaku { początku bloku.

13.1.7 (7) Oczekiwany przecinek lub nawias.

Błąd w wywołaniu wbudowanej funkcji. Został pominięty średnik rozdzielający poszczególne argumenty funkcji lub zapomniano wpisać nawias ')' zamykający listę argumentów.

13.1.8 (8) Oczekiwany nawias zamykający.

Błąd występuje dla funkcji ze stałą liczbą parametrów i oznacza brak nawiasu ')' zamykającego listę argumentów funkcji. Wystąpi również, gdy podamy większą liczbę parametrów niż wymagana przez funkcję.

13.1.9 (9) Zignorowano wyrażenie poza sekcją.

Wszystkie wywołania funkcji oraz podstawienia wartości (oprócz wyrażeń zaczynających się znakiem #) mogą być umieszczone wyłącznie wewnątrz sekcji (procedury) szablonu. Wyrażenia które znajdują się poza sekcją będą zignorowane.

13.1.10 (10) Błąd otwarcia pliku szablonu: ...

Import specjalny nie mógł otworzyć pliku szablonu. Podany plik może nie istnieć, być uszkodzony lub otwarty w trybie wyłączności przez innego użytkownika. Może być również umieszczony w katalogu, do którego użytkownik nie ma żadnych praw dostępu

13.1.11 (11) Nieoczekiwany koniec pliku.

Podczas analizy szablonu osiągnięto koniec pliku, ale nie znaleziono znaku } kończącego rozpoczętą sekcję, lub polecenie blokowe. Oznacza to, że w szablonie jest więcej znaków { rozpoczynających blok wyrażeń, niż znaków } kończących blok.

13.1.12 (12) Brak znaku równości.

W wyrażeniu podstawienia nie podano znaku =. Błąd wystąpi wówczas, gdy na początku linii rozpoznano pole wyjściowe lub zmienną, po której nie było znaku podstawienia =.

13.1.13 (13) Nierozpoznany symbol: ...

Podany symbol nie został rozpoznany jako poprawna nazwa funkcji, nazwa zmiennej lub pola. Należy sprawdzić poprawność zapisu nazwy, albo obecność znaków \$, @ lub [na początku nazwy.

13.1.14 (14) Oczekiwane polecenie proste.

Jako argumentu funkcji lub podstawienia użyto polecenia złożonego (np. IF, SWITCH, WHILE), które nie zwraca wartości i nie może być użyte jako argument.

13.1.15 (15) Nieznany typ polecenia: ...

Błąd wewnętrzny języka szablonów. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o.

13.1.16 (16) Oczekiwany nawias otwierający.

Podano nazwę funkcji wbudowanej, ale po niej nie podano oczekiwanego nawiasu otwierającego '('.

13.1.17 (17) Oczekiwany przecinek.

Podano mniejszą liczbę argumentów funkcji niż jest to wymagane, lub nie podano średnika oddzielającego poszczególne argumenty funkcji.

13.1.18 (19) Pole '...' nie zostało znalezione.

Błąd: Użyto polecenia FIELD, ale pole wyjściowe o podanej nazwie nie istnieje. W nazwach pól wszystkie białe znaki z początku i końca nazwy są ignorowane, a białe znaki w środku wielocłonowej nazwy są traktowane jako jedna spacja. Pola są wyszukiwane bez rozróżniania małych i wielkich liter, a porównanie małych i wielkich liter narodowych (ą-Ą, e-Ę itd.) wykorzystuje bieżącą stronę kodową Windows.

Błąd: Po lewej stronie podstawienia nie ma pola wyjściowego lub zmiennej (np. podano pole wyjściowe)

Ostrzeżenie: Po prawej stronie podstawienia użyto nazwy nieistniejącej zmiennej, pola wyjściowego lub pola wyjściowego. Wynikiem takiego podstawienia będzie zawsze pusty tekst „”.

13.1.19 (20) Argument nie zawiera liczby.

Jako argument funkcji numerycznej podano niepusty tekst nie zawierający liczby (pierwszy znak argumentu jest różny od cyfr lub znaków + lub -).

13.1.20 (21) Wynik jest poza dopuszczalnym zakresem.

Wynik operacji arytmetycznej wykracza poza dopuszczalny rozmiar. Jako błędne przyjęto liczby posiadające więcej niż 20 znaków przed kropką dziesiętną.

13.1.21 (22) Pusta nazwa zmiennej lub definicji.

Po znaku # lub @ nie podano nazwy zmiennej lub definicji, albo użyto znaków niedozwolonych w nazwie zmiennych lub definicji. Pierwszy znak musi być zawsze literą.

13.1.22 (23) Definicja nie może być umieszczona wewnątrz sekcji lub polecenia.

Polecenia globalne, rozpoczynające się znakiem # nie mogą być umieszczane wewnątrz sekcji. Jeśli potrzebujemy zmienić lub utworzyć definicję w trakcie wykonania szablonu, należy użyć polecenia DEF.

13.1.23 (24) Oczekiwana wartość tekstowa.

Jako argument wyrażenia jest oczekiwana bezpośrednia wartość tekstowa. Błąd jest zgłaszany np. jeśli wyrażeniu CASE podamy zmienną zamiast wartości bezpośredniej (tekst w cudzysłowie).

13.1.24 (25) Oczekiwane 'CASE'.

Po poleceniu SWITCH i nawiasie otwierającym blok oczekiwano klauzuli 'CASE'

13.1.25 (26) Błędnie umieszczony znak: ...

Podany znak specjalny (),=[] {} itp. jest niewłaściwie umieszczony i nie spełnia wymogów poprawności składniowej wyrażenia.

13.1.26 (27) Zbyt wiele otwartych szablonów.

Użyto zbyt wiele poleceń #EXECUTE, albo to polecenie otwiera rekurencyjnie ten sam szablon powodując „zapętlenie” szablonów. Poleceniem #EXECUTE Można użyć maksymalnie 10 szablonów.

13.1.27 (27) Nieznane polecenie globalne: ...

Użyto składni polecenia globalnego, ale podany symbol nie został rozpoznany jako polecenie globalne. Obecnie używanymi poleceniami globalnymi są #Open oraz #Execute

13.1.28 (28) Sekcja wyjściowa nie została utworzona.

Próbowano wykonać podstawienie do pola wyjściowego, bez wcześniejszego utworzenia sekcji wyjściowej. Sekcję należy utworzyć podając niepusty drugi argument w poleceniu SECTION albo jawnie wywołując polecenie MAKESEC.

13.1.29 (29) Pominięto sekcję: ...

Podczas wykonania polecenia SECTION z parametrem FIND natrafiono na sekcję o nazwie nie pasującej do poszukiwanej. Sekcja ta została zignorowana (pominięta)

13.1.30 (30) Brak otwartej sekcji wejściowej.

Błąd nr 30 nie jest używany, nie powinien nigdy wystąpić.

13.1.31 (31) Niepoprawna lub niezgodna specyfikacja formatu daty.

Podana data zawiera inne separatory niż podane w specyfikacji polecenia #DATA, lub zawiera dodatkowe znaki różne od cyfr i separatorów.

13.1.32 (32) Dzielenie przez zero.

Jako drugi argument polecenia DIV użyto wartości zero lub pustego stringu

13.1.33 (33) Po znaku + lub - oczekiwana jest liczba.

W wartości bezpośredniej, po znaku liczby natrafiono na znaki różne od spacji lub cyfr.

13.1.34 (34) Polecenie #BASIC może wystąpić tylko raz.

W szablonie użyto polecenia #BASIC więcej niż raz. Po pierwszym użyciu motor języka raportów jest załadowany i nie można ponownie go włączać.

13.1.35 (200) Błąd wewnętrzny, zły numer polecenia: ...

Błąd wewnętrzny języka szablonów. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o.

13.1.36 (201) Zły typ polecenia.

Błąd wewnętrzny języka szablonów. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o..

13.1.37 (202) Błąd wewnętrzny, zły typ argumentu: ...

Błąd wewnętrzny języka szablonów. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o.

13.2 Kody błędów związane z plikiem wejściowym

13.2.1 (300) Nierozpoznana sekcja wejściowa: ...

Postać pliku danych został sparsowany ale nie rozpoznano sekcji wejściowej.

13.2.2 (400) W pliku danych nie odnaleziono końca komentarza blokowego.

W pliku danych napotkano na znak początku komentarza blokowego, ale w poszukiwaniu znaku końca natrafiono na koniec pliku. Oznaczać to może błąd w pliku danych lub niewłaściwą definicję znaków początku oraz końca komentarza blokowego.

13.2.3 (401) W pliku danych znaleziono pole z pustą wartością.

W pliku danych natrafiono na pole, którego wartość jest pusta. Oznaczać to może błędy w pliku danych.

13.2.4 (402) Plik z danymi nie został jeszcze otwarty.

Błąd wewnętrzny procedur importu. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o.

13.2.5 (403) W pliku danych nie ma żadnego wpisu.

W pliku danych nie znaleziono żadnych informacji. Jeśli plik nie jest pusty, należy sprawdzić, czy wszystkie linie w pliku nie zostały potraktowane jako komentarze (sprawdzić definicję komentarza liniowego oraz blokowego).

13.2.6 (404) Plik z danymi niedostępny. Nie istnieje lub jest używany przez kogoś innego.

Nie można otworzyć pliku danych. Podany plik może nie istnieć, być uszkodzony lub otwarty w trybie wyłączności przez innego użytkownika. Może być również umieszczony w katalogu, do którego użytkownik nie ma żadnych praw dostępu

13.2.7 (405) Plik z danymi jest już otwarty.

Błąd wewnętrzny procedur importu. W przypadku wystąpienia należy skontaktować się z firmą Sage sp. z o.o.

13.2.8 (406) Nie podano formatów sekcji i pliku.

W szablonie nie podano specyfikacji formatu pliku wejściowego. Należy użyć predefiniowanego formatu poleceniem #FORMAT, lub zdefiniować format jawnie w szablonie.

13.2.9 (407) Nierozpoznany identyfikator typu pliku w szablonie.

Nazwa formatu pliku podana w szablonie poleceniem #FORMAT nie istnieje (nie jest zdefiniowana w pliku IMPEXP.INI). Należy sprawdzić prawidłowość pisowni nazwy formatu lub zdefiniować nowy format w pliku szablonu.

13.2.10 (408) Niepoprawny element formatu sekcji lub pliku w szablonie.

Komunikat 408 nie jest używany i nie powinien wystąpić.

13.2.11 (409) Znaleziono w pliku element nie odpowiadający żadnemu wzorcowi.

W pliku danych natrafiono na dane, których nie można było dopasować do żadnej definicji formatu. Oznaczać to może błędy w pliku (uszkodzenie lub błędne wpisy), użycie niewłaściwej nazwy formatu w szablonie lub przypadkowe wczytanie niewłaściwego pliku.

13.2.12 (410) Zbyt dużo zagnieżdżonych podsekcji.

Podczas wczytywania jednej sekcji danych wejściowych stwierdzono zbyt dużą liczbę podsekcji danych. Oznaczać to może błędnie zdefiniowany własny format danych w szablonie (potraktowanie kolejnych sekcji danych jako podsekcji). W pliku danych może być najwyżej 20 poziomów zagłębień podsekcji danych

13.2.13 (411) Nie podano listy szerokości pól formatu CDF.

W specyfikacji formatu CDF (pól o stałej szerokości) nie podano ilości znaków w poszczególnych polach (liczb oddzielonych przecinkami).

13.2.14 (412) Błąd w wyrażeniu regularnym.

Specyfikacja formatu zdefiniowanego przez użytkownika zawiera niepoprawne definicje wyrażeń regularnych. Pełny opis wyrażeń regularnych znajduje się w pliku pomocy do języka raportów (AMBASFK.CHM)

13.2.15 (413) Niepoprawne formaty sekcji w szablonie.

Zdefiniowany przez użytkownika własny format danych jest niekompletny, np. brakuje w nim definicji wartości, początku albo końca sekcji.

13.3 Kody błędów związane z kontrolą poprawności

13.3.1 (600) Nieprawidłowa identyfikacja zapisów na konto kasy.

Dla dokumentu typu **Raport Kasowy** liczba zapisów na konto kasy nie zgadza się z liczbą zapisów na konta przeciwstawne. Dotyczy to wyłącznie zapisów podstawowych (nie dotyczy księgowañ równoległych). Przyczyny mogą być następujące:

użycie w danych wejściowych zapisów rozbitych po jednej stronie

brak numeru konta kasy w zapisach

brak numeru konta kasy w sekcji dokumentu oraz niewypełnione pole *\$KontoKasy w zapisach*

niezgodność numeru konta kasy w sekcji zapisów i dokumentu (jeśli zapisy nie mają wypełnionego pola *\$KontoKasy*)

13.3.2 (601) W definicji użytego rejestru nie ma wybranej stawki ...

W definicji rejestru VAT użytego dla importowanego dokumentu nie ma zdefiniowanej takiej stawki procentowej VAT, jaka jest podana w danych importowanego dokumentu. Błąd może wystąpić tylko wtedy, gdy w danych wejściowych są jawnie podane wartości procentowe dla poszczególnych stawek VAT (w polach *\$stawka1 ... \$stawka4*).

13.3.3 (602) Zapis nr ... Nie odnaleziono transakcji do rozliczenia. Rozliczenie zastąpiono nową transakcją.

Podczas importu rozliczenia (związanego z zapisem o podanym numerze kolejnym) nie znaleziono właściwej transakcji do rozliczenia. Przyczyny mogą być następujące:

Nie ma właściwej transakcji do rozliczenia (nie wprowadzono jeszcze dokumentu rozpoczynającego transakcję lub nie wypełniono dla niego rozrachunków)

konto zapisu (i rozliczenia) nie zgadza się z kontem transakcji do rozliczenia (błąd w numerze konta)

numer własny dokumentu rozpoczynającego transakcję jest inny niż numer podany w rozliczeniu.

Transakcja do rozliczenia została już rozliczona przez inny dokument (np. dokument rozliczający został wprowadzony ręcznie lub powtórnie importujemy ten sam dokument)

13.3.4 (603) Niezgodność rodzaju zakupów (ABC) pomiędzy danymi rejestru i jego definicją.

W danych importowanych dla rejestru zakupu określono rodzaj zakupu (A,B lub C), którego nie ma w definicji rejestru VAT użytego dla importowanego dokumentu. Albo wybrano niewłaściwą definicję rejestru VAT dla importowanego dokumentu, albo w danych wejściowych nie został prawidłowo określony rodzaj zakupu.

13.3.5 (604) Brak zapisów w dokumencie.

W danych importowanego dokumentu brakuje zapisów na konta. Przyczyny mogą być następujące:

niepoprawny lub uszkodzony (ucięty) plik danych

błędnie skonstruowany szablon importu specjalnego lub zmieniony sposób identyfikowania sekcji zapisów (uwaga na wielkość liter w nazwie podsekcji)

dokument typu faktura VAT tylko z danymi rejestru VAT (zapisy generowane automatycznie na podstawie rejestrów) został połączony z niewłaściwym typem dokumentu w programie (bez rejestrów VAT)

13.3.6 (605) Nie odnaleziono transakcji do rozliczenia. Rozliczenie zastąpiono nową transakcją.

Podczas importu rozliczenia (związanego z kontrahentem dla faktur VAT i rachunków uproszczonych) nie znaleziono właściwej transakcji do rozliczenia. Przyczyny mogą być następujące:

Nie ma właściwej transakcji do rozliczenia (nie wprowadzono jeszcze dokumentu rozpoczynającego transakcję lub nie wypełniono dla niego rozrachunków)

konto kontrahenta na dokumencie (i rozliczenia) nie zgadza się z kontem transakcji do rozliczenia (błąd w numerze konta)

numer własny dokumentu rozpoczynającego transakcję jest inny niż numer podany w rozliczeniu.

Transakcja do rozliczenia została już rozliczona przez inny dokument (np. dokument rozliczający został wprowadzony ręcznie lub powtórnie importujemy ten sam dokument).

13.3.7 (607) Dokument: ... wystąpił błąd podczas zapisywania.

Podczas zapisywania dokumentu o podanym numerze własnym wystąpił błąd obsługi bazy danych i dokument nie został zapisany. Należy sprawdzić i usunąć przyczynę wystąpienia błędu.

13.3.8 (608) Brak definicji rejestru użytego w uzgodnieniu.

Podczas importu dokumentu zawierającego rejestr VAT stwierdzono, że nie istnieje definicja rejestru użyta podczas poprzednich sesji importu dla tego typu dokumentu. Przyczyny są następujące:

usunięto ręcznie tę definicję rejestru

odtworzono kopię bezpieczeństwa z okresu przed utworzeniem tej definicji rejestru

W obydwu przypadkach należy otworzyć okno uzgodnień dla danego profilu i usunąć błędne połączenie definicji rejestrów.

13.3.9 (609) Błąd wykonania procedury AMBASIC.

Błąd zarezerwowany dla przyszłych wersji importu specjalnego.

13.3.10 (610) Brak pola lub pusta wartość pola 'klucz' w uzgadnianej sekcji.

Podczas importu specjalnego nie została prawidłowo określona wartość unikalnego identyfikatora danych (pola \$klucz). Przyczyny mogą być następujące:

w szablonie nie podstawiono wartości do pola \$klucz

w danych wejściowych pole wybrane jako unikalny identyfikator jest puste

Błąd ten może wystąpić dla sekcji kartotek, definicji rejestrów oraz definicji dokumentów

13.3.11 (611) Nie znaleziono zapisu odpowiadającego sekcji rozrachunków.

Podczas importu dokumentu z rozrachunkami związanymi z całym dokumentem (WAŻNE – są to wyłącznie faktury, rachunki uproszczone - nie dotyczy to więc korekt) nie udało się odnaleźć zapisu na konto rozrachunkowe (konto nie istnieje, nie zostało podane, nie jest rozrachunkowe, lub podane jest po nieprawidłowej stronie). Dla pozostałych typów dokumentów sekcja rozrachunków powinna być umieszczona w sekcji zapisu a nie dokumentu. A więc przyczyny mogą być dwojakie:

- sekcja rozrachunków jest w sekcji dokumentu a nie zapisu(błędnie) – co równoważne jest także ustawieniu wartości [idDlaRozliczen] na -1,

- sekcja rozrachunków jest umiejscowiona poprawnie (i wtedy zapis dla celów rozrachunków identyfikowany jest poprzez wartość pola \$idDlaRozliczen) – gdy nie odnaleziono zapisu o podanym identyfikatorze lub konto tego zapisu jest nieprawidłowe (nie istnieje lub nie jest rozrachunkowe)

13.3.12 (612) Nieistniejące lub nieprawidłowe konto dla rozrachunków. Rozrachunki zostały zignorowane.

Konto kontrahenta (dla faktur VAT i rachunków uproszczonych) nie jest prawidłowe, tzn. nie jest podane w ogóle, nie jest rozrachunkowe lub nie jest ostatnim poziomem analityki.

13.3.13 (613) Zapis nr ... Nieistniejące lub nieprawidłowe konto dla rozrachunków. Rozrachunki zostały zignorowane.

Konto zapisu wchodzącego do rozrachunków nie jest prawidłowe, tzn. nie jest podane w ogóle, nie jest rozrachunkowe lub nie jest ostatnim poziomem analityki.

13.3.14 (614) Rejestr ... ma okres przyszłoroczny, jednak w przyszłym roku nie istnieje zgodna definicja rejestru.

Błąd związany z poprzednimi wersjami programu. Nie uzyskiwany od wersji 5.10.

13.3.15 (615) Rejestr VAT nie ma określonego rodzaju zakupu ABC

Dla rejestru typu zakup lub import nie został określony typ ze względu na zakup

13.3.16 (616) W rejestrze VAT nie ma podanej stawki VAT.

W danych sekcji rejestru podano informacje o stawce lecz nie określono lub źle określono jej wartość.

13.3.17 (617) Niekompletne dane rejestru VAT.

Nie zidentyfikowano danych dotyczących rejestru VAT lub podano je w sposób niekompletny, tak że w efekcie niemożliwe było określenie rejestru, np. pomylenie rodzaju rejestru z nazwami pól dla starszego sposobu rozpoznania rejestru VAT.

13.3.18 (618) Pominięte nie pełne informacje o rozliczeniu w różnych walutach

Podczas importowania rozrachunków różno walutowych (kiedy zapisy mające powodować rozliczenia są w innych walutach niż transakcje) nie wyspecyfikowano wszystkich pól koniecznych dla skonfigurowania rozliczenia: kursWO, walutaObca, rozliczonoWO.

13.3.19 (619) do (623)

Błędy wskazujące na niezgodność przeliczeń walutowych przy określaniu rozliczeń różnowalutowych

13.3.20 (624) wymuszenie daty okresu

Dla importowanego dokumentu data okresu ustawiona była poza obecny rok. Okres dokumentu został zmieniony na ostatni w bieżącym roku.

14 Odpowiedzi na najczęściej zadawane pytania

Jakie typy dokumentów księgowych mogą być wczytywane przez import specjalny?

Wczytywane mogą być wszystkie typy dokumentów występujące w programie Finanse i Księgowość. Podczas importu dokumentów, można również wczytywać informacje do kartotek programu FK.

Czy cechy dokumentów i zapisów mogą być wczytywane przez import specjalny?

Nie. Obecna wersja importu specjalnego nie ma możliwości importowania cech ani nadawania nowych cech dokumentom oraz zapisom

Skąd import specjalny wie jaki typ dokumentu w FK odpowiada dokumentowi importowanemu?

O powiązaniu pomiędzy dokumentami importowanymi a typami dokumentów wzorcowych decyduje użytkownik podczas importu. W szablonie należy ustalić dla każdego typu dokumentu importowanego inną wartość pola \$klucz. Pojawienie się nowej wartości pola \$klucz w sekcji dokumentu powoduje otwarcie okna uzgodnień dokumentów, w którym użytkownik dokonuje ręcznego połączenia typów dokumentów. Aby poprawnie skojarzyć typy dokumentów nadawca powinien ustawić w danych pole 'ObslugujJak' na jeden z wzorcowych typów dokumentów. Wtedy podczas importu użytkownik będzie wiedział jakiego typu dokumentu oczekiwał nadawca.

Jak wypełniać pole „ObslugujJak” w formacie AMS?

Pole to jest sugestią dla użytkownika, z jakim typem wzorcowym powinien być połączony dokument importowany. Pole to może pozostać puste, lub zawierać symbol dokumentu z programu FK, np. „DP”, „RK”, „FVZ” itd. Pozostawienie pustego pola może powodować trudności w prawidłowym uzgodnieniu – użytkownik importujący dane nie będzie miał przesłanek do podjęcia decyzji jaki typ dokumentu należałoby wybrać.

Czy pole \$tresc może zawierać dowolny tekst wpisany przez użytkownika

Tak, to pole jest dowolnym opisem dokumentu i nie ma żadnego szczególnego znaczenia dla programu, to znaczy zawartość tego pola nie ma żadnego wpływu na przebieg importu dokumentów. Pole to może pozostać również puste.

Czy import specjalny umożliwia wczytywanie pozycji słowników?

Nie. Obecna wersja nie importuje pozycji słowników

Czy w polu \$dataOper może być wpisana data dokumentu

Dla dokumentów sprzedaży data operacji gospodarczej jest zwykle taka sama jak data dokumentu. Natomiast dla dokumentów zakupu jest to zwykle data wpływu dokumentu.

Co to są zakupy typu A B oraz C?

Dla dokumentów zakupu można ustalić dla jakiego rodzaju sprzedaży się ten zakup odnosi. Do ustalenia tego rodzaju służy pole \$abc, które należy ustawić na odpowiednią wartość:

A - zakupy związane ze sprzedażą opodatkowaną

- B - zakupy związane ze sprzedażą zwolnioną
- C - zakupy związane ze sprzedażą opodatkowaną i zwolnioną (mieszana)

Czy pola tekstowe w pliku danych mogą być dowolnej długości?

tak, pola danych w pliku mogą być dowolnej długości, to znaczy mogą być dłuższe niż odpowiadające im pola danych w bazie danych programu (i na formatkach wprowadzania dokumentów lub np. kontrahentów). Oczywiście wszystkie pola ostatecznie będą obcinane do rozmiarów określonych pól tekstowych w kartotekach lub na dokumencie, bez żadnego ostrzeżenia.

Czy pole \$ulica może zawierać dodatkowo numer domu i mieszkania, a pola numerDomu i numerMieszk pozostać puste?

Tak, pola „ulica” oraz „numer domu” w kartotece kontrahentów są wyłącznie polami informacyjnymi i umieszczenie numeru domu w jednym polu razem z ulicą nie przeszkadza w normalnej eksploatacji programu, w tym w sporządzaniu wydruków zawierających informacje o kontrahencie (np. przelewy, potwierdzenia sald)

Czy w pliku danych muszą być zawarte poszczególne zapisy księgowe z dekretacjami, czy wystarczą same rejestry VAT?

Poszczególne zapisy księgowe są wymagane dla dokumentów następujących typów

DP - dokument prosty

WB - wyciąg bankowy

RK – raport kasowy

Dla faktur VAT oraz rachunków uproszczonych wystarczy podanie o sumarycznych kwotach dla poszczególnych stawek VAT oraz rodzajów zakupu (A,B oraz C). Procedury importu specjalnego tworzą w takim wypadku automatyczne dekretacje, analogicznie jak dla okna wprowadzania dokumentów, w chwili przejścia od wypełnionego rejestru VAT do formatki dokumentu. W sekcji wyjściowej „rejestr” można dodatkowo określić „wzorcowe” numery kont odpowiednio dla kwoty netto, brutto oraz VAT.

W jaki sposób zaklasyfikować fakturę zakupu do określonego rodzaju sprzedaży?

Dla dokumentów zakupu można ustalić dla jakiego rodzaju sprzedaży się ten zakup odnosi. Do ustalenia tego rodzaju służy pole \$abc, które należy ustawić na odpowiednią wartość:

- 1 zakupy związane ze sprzedażą opodatkowaną (A)
- 2 zakupy związane ze sprzedażą zwolnioną (B)
- 4 zakupy związane ze sprzedażą opodatkowaną i zwolnioną (C)

Czy nazwa pliku z danymi jest dowolna?

Nazwę pliku lub maskę plików określa się w profilu importu i może być to dowolna nazwa. Jednakże jeśli dane importowane składają się z wielu plików, a szablon importu zawiera polecenia #execute do wczytania tych plików, wówczas nazwy tych „dodatkowych plików” muszą zostać określone na stałe dla konkretnego szablonu.

Czy można uzyskać telefon kontaktowy lub e-mail bezpośrednio do autorów importu specjalnego ?

Nie, wszystkie sprawy związane z importem specjalnym powinny być zgłaszane do Działu Wsparcia Technicznego Produktu lub do Działu Konsultacji i Wdrożeń.

W jaki sposób można przetestować własne pliki wejściowe i szablon importu?

Przeprowadzając import tych danych, najlepiej do firmy testowej lub zupełnie nowo utworzonej firmy. Pomocą w sprawdzeniu poprawności importu jest debugger importu, uaktywniany przez sekwencję klawiszy Ctrl Alt I, np. na oknie z listą plików oczekujących na wczytanie.

Jak mogę sprawdzić jakie są limity poszczególnych pól danych?

Wszystkie limity pól danych w imporcie specjalnym są analogiczne jak limity odpowiednich pól na formatkach wprowadzania danych (w kartotekach i oknach wprowadzania dokumentów).

Czy w formacie AM stawka pełna jest zawsze 22% a stawka obniżona 7%?

Format AM został stworzony z myślą o programie Finanse i Księgowość 2.0, w którym występowały tylko dwie stawki niezerowe: 22 oraz 7%. W pliku można wpisać dowolne inne stawki, stopy procentowe dla tych stawek zawarte są w ostatnich dwóch polach sekcji „VAT”. Ze względu jednak na znaczne rozszerzenie informacji w rejestrach i dokumentach w kolejnych wersjach programu, stosowanie tego formatu nie jest zalecane.

Czy muszę wypełniać pole \$klucz dla sekcji Dokument, rejestr i Kontrahent?

Tak, tylko dzięki informacji zapisanej w tym polu import specjalny pamięta dokonane połączenia pomiędzy dokumentami w pliku oraz typami dokumentów w FK
rodzajami rejestrów w pliku i definicjami rejestrów w FK
powiązaniem pomiędzy kontrahentami w pliku i kontrahentami w kartotece kontrahentów FK

Czy można wprowadzić dane do kartoteki kontrahentów incydentalnych?

Tak, w tym celu zamiast sekcji wyjściowej „Kontrahenci” należy utworzyć sekcję wyjściową „Incydentalni”

Czy wewnątrz szablonu jest dostępna jakakolwiek informacja o profilu dla którego szablon został uruchomiony?

Od wersji 18.20

Można skorzystać z informacji o nazwie profilu. `Def("NazwaProfilu")` w instrukcjach szablonu.

Wcześniej

Poprzednie wersje importu specjalnego nie udostępniały żadnych informacji z profilu.

Gdzie wprowadzamy kwotę z dokumentu nie idącą do rejestru VAT?

Jest to tak zwana kwota poza rejestrem VAT odnotowywana na dokumentach typu faktura, która faktycznie nie ma odniesienia do kwot rejestrowanych w rejestrze VAT związana np. z opakowaniami zwrotnymi.

Specyfikujemy w sekcji wyjściowej „Dokument” poprzez podstawienie pola \$KwotaPozaRej, np.

\$kwotaPozaRej=[kwotaPozaRej], gdzie kwotaPozaRej jest w pliku danych.

Czy do katalogu archiwum importu specjalnego trafiają pliki zawierające dokonane uzgodnienia ?

Nie, do archiwum trafiają wyłącznie pliki danych wejściowych

Czy archiwum plików wejściowych oraz pliki zawierające uzgodnienia są archiwizowane archiwizacją systemową?

W systemie Symfonia dane te nie są archiwizowane. W Systemie Zarządzania Forte archiwizowane są profile oraz dokonane uzgodnienia.

Czy poprzez import specjalny można importować tabele kursów walut?

Nie, symbol waluty oraz kurs podajemy bezpośrednio w sekcji „dokument” oraz „zapis”

Czy wczytując poprzez import specjalny dane do kartotek podmiotów można ustawiać/pobierać znacznik podmiotu?

Tak, aktualne wersje programu udostępniają w interfejsie podmiotów atrybut [znacznik] , który stosownie do potrzeb może być wczytywany – wówczas będzie identyczny ze znacznikiem u nadawcy danych, ale można też ignorować jego ustawienie lub stosownie do wartości decydować lub też podejmować inną akcję.