

Obiekt Integracji

1	Wstęp	10
1.1	INFORMACJE OGÓLNE.....	10
1.2	SŁOWNIK POJĘĆ	10
1.3	INFORMACJE O ZMIANACH W DOKUMENCIE.....	10
2	Informacje ogólne	11
2.1	PRZEZNACZENIE OBIEKTU	11
3	Opis interfejsów	11
3.1	POLA DANYCH.....	11
3.2	ELEMENTY WSPÓLNE.....	12
3.2.1.1	FieldName.....	12
3.2.1.2	FieldSize	12
3.2.1.3	FieldType	12
3.2.1.4	FieldDec	13
3.2.1.5	FieldCount.....	13
3.3	ITGINFO.....	13
3.3.1	WŁAŚCIWOŚCI	13
3.3.1.1	Info	13
3.3.1.2	MajorVersion	15
3.3.1.3	Description	15
3.3.1.4	Version	15
3.3.1.5	Extension	15
3.4	BTDATABASE.....	15
3.4.1	WŁAŚCIWOŚCI	16
3.4.1.1	FirmalInfo	16
3.4.1.2	ErrorMode	16
3.4.1.3	LastError	16
3.4.1.4	ErrorDesc	17
3.4.1.5	Version	17
3.4.1.6	UserName	17
3.4.1.7	CurrentDate.....	17
3.4.2	METODY	18
3.4.2.1	Check	18
3.4.2.2	Open	18
3.4.2.3	IsOpen.....	19
3.4.2.4	IsUnifiedLogon	19
3.4.2.5	BeginTransaction	19
3.4.2.6	EndTransaction	19
3.4.2.7	AbortTransaction.....	19
3.5	FIRMAINFO	19
3.5.1	METODY	20
3.5.1.1	SetRok	20
3.5.1.2	GetRokByDate	20

3.5.2	WŁAŚCIWOŚCI	20
3.5.2.1	OstatniRok	20
3.5.2.2	RokDenom	20
3.5.2.3	RokCount	21
3.5.2.4	Rok	21
3.5.2.5	CreateDate	21
3.5.3	POZOSTAŁE WŁASNOŚCI UMOŻLIWIAJĄCE ODCZYTANIE NAZWY, DANYCH ADRESOWYCH ORAZ IDENTYFIKACYJNYCH FIRMY.	22
3.5.3.1	Nazwa	22
3.5.3.2	NazwaSkrócona	22
3.5.3.3	Wojewodztwo	22
3.5.3.4	Miejscowosc	22
3.5.3.5	Ulica	22
3.5.3.6	Dom	22
3.5.3.7	Lokal	22
3.5.3.8	KodPocztowy	22
3.5.3.9	Poczta	22
3.5.3.10	NIP	22
3.5.3.11	Regon	22
3.5.3.12	Telefon	22
3.6	ROKINFO	22
3.6.1	WŁAŚCIWOŚCI	22
3.6.1.1	RokId	22
3.6.1.2	Katalog	22
3.6.1.3	PoczątekRoku	23
3.6.1.4	KoniecRoku	23
3.6.1.5	DługośćRoku	23
3.6.1.6	Zamknięty	23
3.6.1.7	WArchiwum	23
3.6.1.8	KsięgowaniaRównoległeAutomatyczne	24
3.6.1.9	Koszty	24
3.7	SYNCROSUBJECT	24
3.7.1	METODY	24
3.7.1.1	Open	24
3.7.1.2	GetRecById	25
3.7.1.3	GetRecByPoz	25
3.7.1.4	GetFirstRec	25
3.7.1.5	GetFirstNewRec	25
3.7.1.6	GetNext	26
3.7.1.7	ClearRecord	26
3.7.1.8	Insert	26
3.7.1.9	Update	26
3.7.1.10	Validate	26
3.7.1.11	CheckIfExists	27
3.7.1.12	GetExistingId	27
3.7.2	WŁAŚCIWOŚCI	27
3.7.2.1	Type	27
3.7.2.2	LastModification	27
3.7.2.3	LastId	28
3.7.2.4	RecordCount	28
3.7.2.5	Value	28
3.7.2.6	FieldName	28
3.7.2.7	FieldType	28
3.7.2.8	FieldSize	29

3.7.2.9	FieldsCount	29
3.7.2.10	BtDatabase	29
3.8	DOKUMENT	29
3.8.1	STRUKTURA OBIEKTU	29
3.8.2	STRUKTURA OBIEKTÓW ZWIĄZANYCH Z DOKUMENTEM KSIĘGOWYM	30
3.8.3	WŁAŚCIWOŚCI	30
3.8.3.1	Value	30
3.8.3.2	Grupa	31
3.8.3.3	Zapis	31
3.8.3.4	RejestrVAT	31
3.8.3.5	Rejestr	32
3.8.3.6	Cecha	32
3.8.3.7	Transakcja	32
3.8.3.8	Rozliczenie	33
3.8.3.9	EOF	33
3.8.4	METODY	33
3.8.4.1	FieldName	33
3.8.4.2	FieldSize	33
3.8.4.3	FieldType	34
3.8.4.4	FieldDec	34
3.8.4.5	FieldCount	34
3.8.4.6	Open	34
3.8.4.7	MoveFirst	35
3.8.4.8	MoveNext	35
3.8.4.9	Insert	35
3.8.4.10	Delete	35
3.8.4.11	Update	36
3.8.4.12	Validate	36
3.8.4.13	Clear	37
3.8.4.14	GetByID	37
3.8.4.15	GetByNR	37
3.8.4.16	GetByGuid	37
3.8.5	ZDARZENIA	38
3.9	ZAPIS	38
3.9.1	OPIS	38
3.9.2	ZASADY TWORZENIA WIERSZY W TABELI ZAPISÓW	38
3.9.2.1	Pojedyncza pozycja w dokumencie	39
3.9.2.2	Pojedyncza pozycja w dokumencie z rozbiciem kwoty po stronie Ma	39
3.9.2.3	Pojedyncza pozycja w dokumencie z rozbiciem kwoty po stronie Wn	40
3.9.2.4	Trzy pozycje różnych typów na jednym dokumencie	40
3.9.3	KSIĘGOWANIA RÓWNOLEGŁE	40
3.9.4	KSIĘGOWANIA RÓWNOLEGŁE AUTOMATYCZNE	41
3.9.5	KSIĘGOWANIA RÓWNOLEGŁE ZBIORCZE	42
3.9.6	KSIĘGOWANIA RÓWNOLEGŁE ZBIORCZE AUTOMATYCZNE	42
3.9.7	WŁAŚCIWOŚCI	42
3.9.7.1	Cecha	42
3.9.7.2	Transakcja	42
3.9.7.3	Rozliczenie	43
3.9.7.4	EOF	43
3.9.7.5	Konto	43

3.9.7.6	Parent.....	44
3.9.7.7	Value	44
3.9.8	METODY	45
3.9.8.1	FieldName.....	45
3.9.8.2	FieldSize	45
3.9.8.3	FieldType	45
3.9.8.4	FieldDec	45
3.9.8.5	FieldCount.....	45
3.9.8.6	MoveFirst	45
3.9.8.7	MoveNext	46
3.9.8.8	Insert	46
3.9.8.9	Delete	46
3.9.9	ZDARZENIA	46
3.10	GRUPA	46
3.10.1	OPIS	46
3.10.1.1	Pojedyncza pozycja w dokumencie z rozbiem kwoty po stronie Wn	47
3.10.1.2	Pojedyncza pozycja w dokumencie z rozbiem kwoty po stronie Ma.....	47
3.10.2	WŁAŚCIWOŚCI	48
3.10.2.1	Parent.....	48
3.10.2.2	Rownolegle	48
3.10.2.3	Rozbicie	49
3.10.2.4	Zapis	50
3.10.3	METODY	50
3.10.3.1	Insert	50
3.10.4	ZDARZENIA	51
3.11	CECHA	51
3.11.1	OPIS	51
3.11.2	WŁAŚCIWOŚCI	51
3.11.2.1	Value	51
3.11.3	METODY	51
3.11.3.1	FieldName.....	51
3.11.3.2	FieldSize	51
3.11.3.3	FieldType	51
3.11.3.4	FieldDec	52
3.11.3.5	FieldCount.....	52
3.11.3.6	MoveFirst	52
3.11.3.7	MoveNext	52
3.11.3.8	Insert	52
3.11.4	ZDARZENIA	53
3.12	TRANSAKCJA	53
3.12.1	OPIS	53
3.12.2	WŁAŚCIWOŚCI	53
3.12.2.1	Value	53
3.12.3	METODY	54
3.12.3.1	FieldName.....	54
3.12.3.2	FieldSize	54
3.12.3.3	FieldType	54

3.12.3.4	FieldDec	54
3.12.3.5	FieldCount.....	54
3.12.3.6	Insert	54
3.12.4	ZDARZENIA	55
3.13	ROZLICZENIE	55
3.13.1	OPIS	55
3.13.2	WŁAŚCIWOŚCI	55
3.13.2.1	Value	55
3.13.3	METODY	56
3.13.3.1	FieldName.....	56
3.13.3.2	FieldSize	56
3.13.3.3	FieldType	56
3.13.3.4	FieldDec	56
3.13.3.5	FieldCount.....	56
3.13.3.6	MoveFirst	57
3.13.3.7	MoveNext	57
3.13.3.8	Insert	57
3.13.4	ZDARZENIA	57
3.14	REJESTRVAT	57
3.14.1	OPIS	57
3.14.2	WŁAŚCIWOŚCI	58
3.14.2.1	EOF	58
3.14.2.2	Parent.....	58
3.14.2.3	Value	59
3.14.2.4	PozycjaVAT.....	59
3.14.2.5	Value	59
3.14.3	METODY	60
3.14.3.1	Open	60
3.14.3.2	FieldName.....	60
3.14.3.3	FieldSize	61
3.14.3.4	FieldType	61
3.14.3.5	FieldDec	61
3.14.3.6	FieldCount.....	61
3.14.3.7	MoveFirst	61
3.14.3.8	MoveNext	61
3.14.3.9	Insert	62
3.14.3.10	Delete.....	62
3.14.4	ZDARZENIA	62
3.15	PREJESTR (UWAGA).....	62
3.15.1	OPIS	62
3.15.2	WŁAŚCIWOŚCI	63
3.15.2.1	EOF	63
3.15.2.2	Parent.....	63
3.15.2.3	Value	63
3.15.3	METODY	64
3.15.3.1	FieldName.....	64
3.15.3.2	FieldSize	64
3.15.3.3	FieldType	64

3.15.3.4	FieldDec	64
3.15.3.5	FieldCount.....	64
3.15.3.6	MoveFirst	64
3.15.3.7	MoveNext	64
3.15.3.8	Insert	65
3.15.3.9	Delete.....	65
3.15.4	ZDARZENIA	65
3.16	DEFDOKUM	65
3.16.1	OPIS	65
3.16.2	WŁAŚCIWOŚCI	65
3.16.2.1	FieldCount.....	65
3.16.2.2	Value	66
3.16.3	METODY	66
3.16.3.1	Open	66
3.16.3.2	MoveFirst	66
3.16.3.3	MoveNext	67
3.16.3.4	MoveTo	67
3.16.3.5	FieldName.....	67
3.16.3.6	FieldSize	67
3.16.3.7	FieldType	67
3.16.3.8	FieldDec	68
3.16.4	ZDARZENIA	68
3.17	DEFREJVAT	68
3.17.1	OPIS	68
3.17.2	WŁAŚCIWOŚCI	68
3.17.2.1	FieldCount.....	68
3.17.2.2	Value	68
3.17.3	METODY	68
3.17.3.1	Open	68
3.17.3.2	MoveFirst	69
3.17.3.3	MoveNext	69
3.17.3.4	MoveTo	69
3.17.3.5	FieldName.....	70
3.17.3.6	FieldSize	70
3.17.3.7	FieldType	70
3.17.3.8	FieldDec	70
3.17.4	ZDARZENIA	70
3.18	DEFREJ (UWAGA)	70
3.18.1	OPIS	70
3.18.2	WŁAŚCIWOŚCI	70
3.18.2.1	FieldCount.....	70
3.18.2.2	Value	70
3.18.3	METODY	71
3.18.3.1	Open	71
3.18.3.2	MoveFirst	71
3.18.3.3	MoveNext	71
3.18.3.4	MoveTo	72
3.18.3.5	FieldName.....	72

3.18.3.6	FieldSize	72
3.18.3.7	FieldType	72
3.18.3.8	FieldDec	72
3.18.4	ZDARZENIA	72
3.19	PLANKONT	73
3.19.1	OPIS	73
3.19.2	WŁAŚCIWOŚCI	73
3.19.2.1	FieldCount.....	73
3.19.2.2	Value	73
3.19.2.3	Listek	73
3.19.2.4	Rozrachunkowe	73
3.19.3	METODY	74
3.19.3.1	Open	74
3.19.3.2	MoveFirst	74
3.19.3.3	MoveNext	74
3.19.3.4	MoveTo	75
3.19.3.5	FieldName.....	75
3.19.3.6	FieldSize	75
3.19.3.7	FieldType	75
3.19.3.8	FieldDec	75
3.19.4	ZDARZENIA	75
3.20	ROZRACHUNKI	75
3.20.1	OPIS	75
3.20.2	WŁAŚCIWOŚCI	75
3.20.2.1	BtDatabase	75
3.20.2.2	Value	76
3.20.2.3	LastId	77
3.20.2.4	LastModification	77
3.20.2.5	SimpleInfo	77
3.20.3	PARAMETRY WYSZUKIWANIA ROZRACHUNKÓW	77
3.20.4	METODY	78
3.20.4.1	FieldCount, FieldName, FieldSize, FieldType	78
3.20.4.2	MoveFirst, MoveNext, MovePrev, MoveLast	78
3.20.4.3	Open	78
3.20.4.4	ClearParams	78
3.20.4.5	ClearNewParams	79
3.20.4.6	ValidateParams.....	79
3.20.4.7	OrderBy, GroupBy.....	79
3.20.4.8	Load	79
3.20.4.9	LoadNew	79
3.21	ROZLICZENIA	80
3.21.1	OPIS	80
3.21.2	WŁAŚCIWOŚCI	80
3.21.2.1	Value	80
3.21.2.2	NaDzien	81
3.21.3	METODY	81
3.21.3.1	FieldCount.....	81
3.21.3.2	FieldName.....	81

3.21.3.3	FieldSize	81
3.21.3.4	FieldType	81
3.21.3.5	MoveFirst, MoveNext	81
3.21.3.6	Open	81
3.22	DOCERRORS	82
3.22.1	OPIS	82
3.22.2	WŁAŚCIWOŚCI	82
3.22.2.1	Errors	82
3.22.2.2	Description	82
3.22.3	METODY	82
3.22.3.1	Count	82
3.22.3.2	Type	82
3.23	INFOKONTA	83
3.23.1	OPIS	83
3.23.2	WŁAŚCIWOŚCI	83
3.23.2.1	Typ	83
3.23.2.2	Podtyp	83
3.23.2.3	Nazwa	83
3.23.2.4	Numer	83
3.23.2.5	Flagi	83
3.23.2.6	Listek	84
3.23.2.7	ZBuforem	84
3.23.2.8	Ma	84
3.23.2.9	Wn	84
3.23.3	METODY	84
3.23.3.1	Pierwsze	84
3.23.3.2	Kolejne	85
3.23.3.3	PierwszeOd	85
3.23.3.4	IdPK	85
3.23.3.5	Info	85
3.23.3.6	Konto	85
3.23.3.7	Obroty	85
3.23.3.8	Open	85
3.23.3.9	Persaldo	86
3.23.3.10	Skrot	86
3.23.3.11	Słownik	86
3.24	INFOPOZ	87
3.24.1	OPIS	87
3.24.2	WŁAŚCIWOŚCI	87
3.24.2.1	BO	87
3.24.2.2	Mc(int okres)	87
3.24.2.3	Cur	87
3.25	POZYCJAVAT	87
3.25.1	OPIS	87
3.25.2	WŁASNOŚCI	87
3.25.2.1	Value	87
3.25.2.2	Parent	88
3.25.3	METODY	89

3.25.3.1	FieldName.....	89
3.25.3.2	FieldSize	89
3.25.3.3	FieldType	89
3.25.3.4	FieldDec	89
3.25.3.5	FieldCount.....	89
3.25.3.6	MoveFirst	89
3.25.3.7	MoveNext	89
3.25.3.8	Insert	90
3.25.3.9	MoveKursVAT	90
3.25.3.10	MoveKursCITPIT.....	90
3.25.4	ZDARZENIA	91
4	Sposób użycia obiektu dokumentu.....	92
4.1	KOLEJNOŚĆ OPERACJI DLA WPROWADZANIA DOKUMENTÓW	92

1 Wstęp

1.1 Informacje ogólne

Wersja publikowanego dokumentu to 1.27.

Dokument opisuje interfejsy aktualne dla wersji 2016 Systemu ERP oraz 2015 Symfonii.

1.2 Słownik Pojęć

Pojęcie	Opis
Symfonia	Nazwa linii programów pracujących na platformie Btrieve/Pervasive (dawniej premium)
Symfonia ERP	Nazwa linii programów pracujących na platformie MS SQL. Nazwa wcześniejszych wersji tej linii to forte.

1.3 Informacje o zmianach w dokumencie

Wersja 1.26 – aktualizacja opisu do wersji 2015.

Wersja 1.25 – dodano informacje o trybie NoRaiseError property [BtDatabase.ErrorMode](#) .

Wersja 1.22 – uwagi po korekcie (2010.12.10)

W wersji 1.21 – zmiany do wersji v2010.2, uzupełniono opis i przykłady w [IPozycjaVAT](#), dodano rozdziały 3.25.3.9 i 3.25.3.10

W wersji 1.20 – dodano opis IPozycjaVAT, uzupełniono IRejestrVAT o zmiany do v2010.1, przykłady

[IPozycjaVAT](#)

[RejestrVAT.PozycjaVAT](#)

W wersji 1.10 – dodano opis Infokonta, uzupełniono IRozrachunki o zmiany do v2010, przykłady

Wersja 1.06 – drobne poprawki i uzupełnienia, dostosowanie do wersji 2008.

W wersji 1.05 – dokumentu uległy zmianie akapity:

[RejestrVAT](#) – nowy

[DefRejVAT](#) – nowy

[PRejestr \(Uwaga \)](#)

[DefRej \(Uwaga \)](#)

2 Informacje ogólne

2.1 Przeznaczenie obiektu

Obiekt integracji umożliwia bezpośrednią wymianę danych pomiędzy programem Sage Symfonia Finanse i Księgowość, a pozostałymi programami systemu Sage Symfonia, oraz programami i systemami innych producentów.

Obiekt został stworzony w oparciu o technologię COM i może być używany w dowolnym języku programowania i środowisku wspierającym tę technologię, np. C++, .net, Visual Basic, Visual Basic for Applications (Word, Excel), Java Script, VB Script, AMBasic oraz wiele innych.

Obiekt integracji umożliwia dostęp do danych programu Finanse i Księgowość z zachowaniem ustawień systemu użytkowników i haseł. Zapewnia dostęp do:

- Informacji o firmie
- Dokumentów księgowych
- Kartotek
- Planu kont
- Definicji dokumentów i rejestrów
- Rozrachunków
- Obrótów kont (również z buforem)

Niniejszy dokument opisuje interfejsy zarówno dla wersji Symfonia jak i Symfonia ERP. w systemie mogą istnieć jednocześnie zarejestrowane oba obiekty do współpracy z wersją Symfonia i Symfonia ERP. W większości przypadków dostęp do obiektów jest niezależny od platformy baz danych.

UWAGA. Każdy z tych obiektów może jednak współpracować z tylko jedną główną wersją bazy danych (ze względu na podatność kodu na zmiany w bazie danych). Tak więc w celu np. zmiany pracy z bazy w formacie 2014 na bazę w formacie 2015 musimy zarejestrować inny, odpowiedni obiekt przeznaczony do obsługi pożądaney wersji bazy.

3 Opis interfejsów

3.1 Pola danych

Każdy obiekt w interfejsie integracji posiada zestaw pól danych, przechowujących dane biznesowe, które zostały odczytane lub zostaną zapisane do bazy danych programu. w większości przypadków ilość pól danych jak też ich nazwy, typy oraz rozmiary odpowiadają bezpośrednio określonym tablicom w bazie danych programu. w przypadku niektórych obiektów zestaw pól może być rozszerzony lub zupełnie zmieniony w stosunku do odpowiadającej mu tabeli w bazie danych. Przy opisie każdego obiektu znajduje się odniesienie do opisu odpowiedniej tabeli bazy danych, lub umieszczony jest bezpośrednio opis wszystkich pól danych.

Do obsługi pól danych każdy obiekt zawiera zestaw metod i właściwości dostępu do tych danych. Są to następujące metody i właściwości:

- FieldName
- FieldCount
- FieldSize
- FieldDec
- FieldType

- Value

3.2 Elementy wspólne

3.2.1.1 FieldName

Składnia:

```
Function Obiekt.FieldName( Index As integer ) As String
```

Opis:

Metoda zwraca tekstową nazwę pola o podanym indeksie numerycznym (z zakresu od 0 do FieldCount() – 1).

Metoda może być używana do pobrania (iteracji) nazw wszystkich pól danych obiektu.

UWAGA: Numeryczny indeks pola nie może być używany, jako permanentny identyfikator pola. Indeks ten jest stały tylko dla jednej, konkretnej instancji obiektu i nie może być użyty do identyfikacji pól w innych obiektach i innych instancjach tego samego obiektu. Jedynym stałym identyfikatorem pola jest jego nazwa tekstowa.

Przykład:

```
' dla obiektu dokumentu dok prezentuje nazwy pól i ich typy
' analogiczne użycie dla innych typów obiektów
For i = 0 to dok.FieldCount()-1
    MsgBox Cstr(i) + " =" + dok.FieldName(i) + ", typ =" + Cstr(dok.FieldType(i))
Next i
```

3.2.1.2 FieldSize

Składnia:

```
Function Obiekt.FieldSize( Index As Variant) As integer
```

Opis:

Metoda zwraca rozmiar pola (w bajtach) o podanym indeksie tekstowym (nazwie pola) lub numerycznym (z zakresu od 0 do FieldCount() – 1). Dla pól liczbowych określa on zakres przechowywanych liczb, dla pól tekstowych jest to maksymalna liczba znaków, które można wprowadzić do pola powiększona o 1.

UWAGA: Numeryczny indeks pola nie może być używany, jako permanentny identyfikator pola. Indeks ten jest stały tylko dla jednej, konkretnej instancji obiektu i nie może być użyty do identyfikacji pól w innych obiektach i innych instancjach tego samego obiektu. Jedynym stałym identyfikatorem pola jest jego nazwa tekstowa.

Przykład:

```
For i = 0 to dok.FieldCount()-1
    MsgBox Cstr(i) + " =" + Cstr(dok.FieldSize(i))
Next i
```

3.2.1.3 FieldType

Składnia:

```
Function Obiekt.FieldType( Index As Variant) As integer
```

Opis:

Metoda zwraca typ pola o podanym indeksie tekstowym (nazwie pola) lub numerycznym (z zakresu od 0 do FieldCount() – 1).

Zwracana liczba określa typ pola w następujący sposób:

Liczba całkowita	1
------------------	---

Kwota, liczba zmiennoprzecinkowa	2
Data	3
Wartość logiczna	7
Tekst	11

UWAGA: Numeryczny indeks pola nie może być używany, jako permanentny identyfikator pola. Indeks ten jest stały tylko dla jednej, konkretnej instancji obiektu i nie może być użyty do identyfikacji pól w innych obiektach i innych instancjach tego samego obiektu. Jedynym stałym identyfikatorem pola jest jego nazwa tekstowa.

Przykład:

```
For i = 0 to dok.FieldCount()-1
    MsgBox Cstr(i) + " =" + Cstr(dok.FieldType(i))
Next i
```

3.2.1.4 FieldDec

Składnia:

```
Function Obiekt.FieldDec( Index As Variant) As integer
```

Opis:

Metoda zwraca ilość miejsc po przecinku przechowywaną dla pól zmiennoprzecinkowych (pól kwoty). Dla pozostałych typów pól jest to wartość 0.

Domyślnie dla pól kwotowych jest to wartość 2, a dla kursów walut wartość 6.

Przykład:

```
For i = 0 to dok.FieldCount()-1
    MsgBox Cstr(i) + " =" + Cstr(dok.FieldDec(i))
Next i
```

3.2.1.5 FieldCount

Składnia:

```
Function Obiekt.FieldCount() As integer
```

Opis:

Metoda zwraca ilość pól danych dostępnych w danym obiekcie. Jest to równocześnie wartość o 1 większa od indeksu ostatniego pola w kolekcji pól obiektu.

Przykład:

```
For i = 0 to dok.FieldCount()-1
    MsgBox Cstr(i) + " =" + Cstr(dok.FieldDec(i))
Next i
```

3.3 ItgInfo

Obiekt ItgInfo dostarcza informacji o zainstalowanych w systemie wersjach integracji oraz umożliwia kwalifikowany dostęp do tych obiektów będący skutkiem decyzji użytkownika, z którym obiektem chce współpracować.

3.3.1 Właściwości

3.3.1.1 Info

Składnia:

```
ItgInfo.Info(String infoType) As Long
```

Opis:

Właściwość **Info** umożliwia pobranie stanu następujących parametrów informujących o rozszerzeniach, które zostały wprowadzone do obiektu, co wiąże się ze zwiększoną funkcjonalnością lub rozbudową o konfigurowanie. Każdy z tych parametrów (poza VersionCount) może przyjąć wartość 0 lub 1. Wartość 1 informuje, że uruchomiona wersja Integracji obsługuje dany interfejs/rozszerzenie.

Lista tych parametrów:

"Rozrachunki" – czy dana wersja obsługuje rozrachunki

"Dokumenty" – czy dana wersja obsługuje dokumenty

"Subject" – czy dana wersja obsługuje synchronizację podmiotów

"VersionCount" – liczba zainstalowanych wersji obiektu integracji w systemie (co najwyżej dwa)

"RejestrOpen" – oznacza dodatkową funkcjonalność obiektu RejestrVat, który może istnieć samodzielnie. Patrz opis obiektu.

"NAKK" – oznacza rozbudowę kartotek podmiotów (używanych w planie kont) o obsługę atrybutu "pozycja" oraz odczyt rekordów – patrz GetRecByPoz.

"PlanKontSubjectType" – wprowadzony od wersji 2008. Oznacza możliwość odpytania planu kont o typ kartoteki z ostatniego poziomu (tzw. listka). Ze względu na przeniesienie kartotek w wersji forte do zarządzania przez SymfoniaCommon numer kartoteki w planie kont nie ma już stałych wartości. Aby zachować możliwość porównania z istniejącymi wartościami SyncroSubjectType przekazywanymi do interfejsu SyncroSubject wprowadzone został taki nowy atrybut i w nim przechowana jest wartość zgodna z konwencją dotychczasowego numerowania.

"InfoKonta" – czy dana wersja obsługuje pobieranie danych o obrotach kont

"RozrachunkiGuid" – rozszerzenie interfejsu IRozrachunki, o możliwość trwałej identyfikacji rozrachunków, związana również z wprowadzeniem unikalnych identyfikatorów (uid) dla zapisów i dokumentów. Wprowadzona tylko dla forte od wersji 2010, funkcjonalność służy szybkiej synchronizacji rozrachunków i jest zaimplementowana po stronie klienckiej w Handel forte od wersji 2010.

"ePrzelewy" – rozszerzenie wprowadzone od wersji 2013 Symfonii wskazujące na współdziałanie ITG z modulem EDS(RDF). Pozwala na wyszukiwanie rozrachunków po podaniu Guida operacji Bankowej pobranej z modułu ePrzelewy.

„NoRaiseError” – rozszerzenie właściwości BtDatabase.ErrorMode o możliwość ustawienia trybu pracy bez generowania wyjątków, tylko zakończenie operacji z informacją, że był błąd w postaci rezultatu wywołania operacji. Rozszerzenie ErrorModeType o wartość noRaiseError = 3.

"GetByGuid" – rozszerzenie wprowadzone od wersji 2013.1 wskazuje rozszerzenie interfejsu Dokumentu pozwalające na wyszukiwanie dokumentów po polu Guid. Atrybut guid dokumentu jest wspólny z modułami RDF, HMF.

Przykłady Info

```
Dim obInfo as ItgInfo  
obInfo.Info("nakk")      ' wartość 0 lub 1
```

3.3.1.2 MajorVersion

Składnia:

```
ItgInfo.MajorVersion(Long nIndex = -1) As Long
```

Opis:

Właściwość **MajorVersion** umożliwia pobranie głównej wersji interfejsów obiektu. Parametr nIndex określa index obiektu w systemie. Przy zastopowaniu wartości domyślnej otrzymujemy wersję domyślnego obiektu.

Przykład

```
Dim oItgInfo as ItgInfo  
MsgBox Str(oItgInfo.MajorVersion)
```

3.3.1.3 Description

Składnia:

```
ItgInfo.Description(Long nIndex = -1) As String
```

Opis:

Właściwość **Description** umożliwia pobranie tekstowego opisu danej wersji obiektu integracji. Opis ten Umożliwi określenie wersji obiektu przez użytkownika danego rozwiązania.

3.3.1.4 Version

Składnia:

```
ItgInfo.Version(Long nIndex = -1) As String
```

Opis:

Właściwość **Version** umożliwia pobranie tekstowej wersji interfejsów obiektów, którą zawiera wersję minor i major (np. „1.0”, „2.1”).

Parametr nIndex - patrz właściwość **MajorVersion**.

3.3.1.5 Extension

Składnia:

```
ItgInfo.Extension(Long nIndex = -1) As String
```

Opis:

Właściwość **Extension** umożliwia pobranie tekstowej wersji interfejsów obiektów, którą można kwalifikować tworzone obiekty (np. „1”, „2.1”).

Parametr nIndex - patrz właściwość **MajorVersion**.

Przykład

```
Dim oItgInfo as new ItgInfo  
Dim database  
Set database = CreateObject("MXDokFK.BtDatabase" + oItgInfo.Extension())
```

3.4 BtDatabase

Obiekt BtDatabase reprezentuje połączenie z bazą danych firmy programu Finanse i Księgowość.

Otwarcie połączenia z bazą danych programu FK jest niezbędne do pracy ze wszystkimi obiektami modułu integracji, oraz do zapewnienia autoryzowanego dostępu do danych wybranej firmy w programie FK

3.4.1 Właściwości

3.4.1.1 FirmaInfo

Składnia:

```
BtDatabase.FirmaInfo As FirmaInfo
```

Opis:

Właściwość **FirmaInfo** umożliwia pobranie obiektu z danymi firmy określonej przez otwarte połączenie do bazy danych.

Próba pobrania tej właściwości bez wcześniejszego otwarcia połączenia spowoduje zgłoszenie błędu.

Przykład

```
Dim oFirmaInfo as FirmaInfo  
Set oFirmaInfo = database.FirmaInfo
```

3.4.1.2 ErrorMode

Składnia:

```
BtDatabase.ErrorMode As ErrorModeType
```

Opis:

Właściwość umożliwia pobranie i ustawienie trybu obsługi błędów bazy danych.

Możliwe są wartości:

showErrorrrsMode = 1, (wartość domyślna.) – pokazuje dialogi z informacjami o komunikatach bazy danych

silentMode = 2 – Nie są pokazywane okna dialogowe z informacją o błędach. Informacja o błędach jest dostępna w odpowiednich polach obiektów wykonujących operację lub poprzez niezerowe wartości zwracane przez funkcje.

noRaiseError = 3 – nie jest generowany wyjątek o błędnym zakończeniu operacji, tylko rezultat operacji mówi czy operacja zakończyła się sukcesem czy błędem – tryb wprowadzony na użytek języka AmBasic (*nie przerywany jest raport AmBasic w razie błędu operacji, która normalnie generuje wyjątek.*), Tryb ten wprowadzony jest od wersji 2013 forte, 2013.1 Symfonia.

Przykład

```
Dim errMode As ErrorModeType  
errMode = database.ErrorMode  
database.ErrorMode = silentMode
```

3.4.1.3 LastError

Składnia:

```
BtDatabase.LastError As Integer
```

Opis:

Właściwość tylko do odczytu zwracająca kod błędu ostatniej operacji. w przypadku, gdy operacja została zakończona pomyślnie zwracane jest 0.

Przykład

```
Dim nLastError = database.LastError
```


3.4.1.4 ErrorDesc

Składnia:

```
BtDatabase.ErrorDesc As String
```

Opis:

Właściwość tylko do odczytu zwracająca opis błędu odpowiadający kodowi błędu ostatniej operacji. w przypadku, gdy operacja została zakończona pomyślnie zwracane jest pusty string.

Przykład

```
Dim sErrorDesc = database.ErrorDesc
```

3.4.1.5 Version

Składnia:

```
BtDatabase.Version As String
```

Opis:

Właściwość **Version** umożliwia pobranie wersji bazy danych, z jaką pracuje obiekt integracji. w przypadku stwierdzenia nowszej wersji bazy danych niż obsługiwana przez obiekt integracji, należy zainstalować uaktualnioną wersję obiektu integracji z płytki instalacyjnej programu Finanse i Księgowość, lub pobrać uaktualnioną wersję ze stron www.sage.com.pl

Przykład

```
Dim sVersion = database.Version
```

3.4.1.6 UserName

Składnia:

```
BtDatabase.UserName As String
```

Opis:

Własność zwraca nazwę użytkownika programu Finanse i Księgowość.

Własność tylko do odczytu ustawiana przez metodę

Przykład

```
Dim sFKUserName = database.UserName
```

3.4.1.7 CurrentDate

Składnia:

```
BtDatabase.CurrentDate As Date
```

Opis:

Własność określa datę z jaką użytkownik działa na bazie danych programu Finanse i Księgowość.

Własność może być ustawiana w celu zarządzania 'datą dla programu' – warunkuje datę zapisu dokumentu do bazy.

Przykład

```
Dim currDate = database.CurrentDate  
currDate = "2015-09-06" ' arbitralnie  
currDate = Application.WybranaFirma.AktualnaData ' dla CoMa wołanego z  
modułu, ustawienie daty takiej jak w aplikacji FK
```

3.4.2 Metody

3.4.2.1 Check

Składnia:

```
BtDatabase.Check (bstrPath As String, ByRef shortName As String, ByRef Version As String) As Boolean
```

Opis:

Metoda sprawdza czy w określonym katalogu znajdują się dane programu Finanse i Księgowość. Jeśli w danym katalogu nie ma danych firmy lub są one w nieodpowiedniej wersji, funkcja zwraca 0. w przeciwnym przypadku wartość jest różna od zera, a dodatkowo zwracany jest skrót nazwy firmy (poprzez parametr shortName) oraz wersja firmy (poprzez Version) znajdujące się w podanym katalogu.

Przykład

```
Dim version As String, shortName As String  
bValid = database.Check("C:\SYMFONIA\demo_fk", shortName, version)
```

3.4.2.2 Open

Składnia:

```
BtDatabase.Open(bstrPath As String, bstrUser As String, bstrPassword As String)
```

Opis:

Metoda umożliwiająca podłączenie się do firmy. Jako bstrUser przekazywana jest nazwa użytkownika a w bstrPassword hasło. Jeśli w wersji Symfonia ERP chcemy używać zintegrowanego logowania należy podać pustą nazwę użytkownika oraz puste hasło.

W wersji Btrieve parametr bstrPath określa ścieżkę do danych firmy.

W wersji Symfonia ERP parametr bstrPath określa string połączeniowy dla ODBC. Zalecane jest przekazanie trzech parametrów:

"Driver=SQL Server;" – jako określenie driver'a ODBC

"Server=???" – zamiast ??? należy wpisać nazwę serwera

"Database=???" – zamiast ??? należy podać nazwę bazy zawierającej dane programu Sage Symfonia ERP Finanse i Księgowość.

bstrUser – należy przekazać login użytkownika aplikacji (nie bazy danych)

Przykładowe parametry połączenia podane są poniżej.

UWAGA!!! Aby zamknąć połączenie z bazą danych należy zwolnić obiekt bazy danych. Nie ma możliwości zmiany bazy danych na otwartym obiekcie BtDatabase.

Przykład

```
'Dla wersji Symfonii (dawniej premium, Btrieve)  
database.Open "C:\SYMFONIA\demo_fk", "admin", "pazzw0rd"  
'Dla wersji Symfonia ERP  
database.Open "Driver=SQL Server;Server=test;Database=FK_Frm;", "JanKowalski",  
"pazzw0rd" 'gdzie JanKowalski jest użytkownikiem aplikacji  
'Dla wersji Symfonia ERP ze zintegrowanym logowaniem  
database.Open "Driver=SQL Server;Server=test;Database=FK_Frm;", "", ""
```

Użycie zarejestrowanego modułu Integracji pozwala na połączenie tylko z jedną odpowiadającą jej wersją bazy danych. Wersja jest tu rozumiana jako numer 2014.x.x, 2015.x.x, etc. z wersji aplikacji.

3.4.2.3 IsOpen

Składnia:

```
BtDatabase.IsOpen() As Boolean
```

Opis:

Metoda zwraca informację, czy dany obiekt bazy danych jest podłączony do danych FK.

Przykład

```
bOpen = database.IsOpen
```

3.4.2.4 IsUnifiedLogon

Składnia:

```
BtDatabase.IsUnifiedLogon() As Boolean
```

Opis:

Metoda określa czy zostało użyte zintegrowane logowanie

UWAGA: Występuje tylko w wersji Symfonia ERP obiektu integracji.

Przykład

```
bIsUnifiedLogon = database.IsUnifiedLogon
```

3.4.2.5 BeginTransaction

Składnia:

```
BtDatabase.BeginTransaction() As Integer
```

Opis:

Metoda rozpoczyna transakcję na połączeniu do bazy danych.

Przykład

```
Status = database.BeginTransaction
```

3.4.2.6 EndTransaction

Składnia:

```
BtDatabase.EndTransaction() As Integer
```

Opis:

Metoda kończy transakcję na połączeniu.

Przykład

```
Status = database.EndTransaction
```

3.4.2.7 AbortTransaction

Składnia:

```
BtDatabase.AbortTransaction() As Integer
```

Opis:

Metoda unieważnia transakcję na połączeniu.

Przykład

```
Status = database.AbortTransaction
```

3.5 FirmaInfo

Obiekt FirmaInfo umożliwia uzyskanie informacji o danych firmy w programie Finanse i Księgowość.

3.5.1 Metody

3.5.1.1 SetRok

Składnia:

```
FirmaInfo.SetRok (rokId As Integer)
```

Opis:

Metoda umożliwia zmianę roku obrachunkowego, na którym pracujemy. Parametrem jest numer kolejny roku obrachunkowego, założonego w programie FK. Pierwszy rok obrachunkowy ma numer 0.

Przykład

```
firma.SetRok(0)
```

3.5.1.2 GetRokByDate

Składnia:

```
FirmaInfo.GetRokByDate (myData As Data) as Integer
```

Opis:

Metoda umożliwia określenie czy do roku z daną datą możemy wstawiać dokumenty. Jeśli zwrócona wartość jest dodatnia lub 0 to jest to identyfikator otwartego roku obrachunkowego, do którego możemy wstawiać dane. Wartości ujemne są kodami błędów:

- (-1) – Nie znaleziono odpowiedniego roku.
- (-2) – Data późniejsza niż koniec ostatniego roku obrachunkowego.
- (-3) – Istnieje odpowiedni rok, ale jest on zamknięty lub zarchiwizowany.

Przykład

```
rokId = firma.GetRokByDate("2000-10-15")
```

3.5.2 Właściwości

3.5.2.1 OstatniRok

Składnia:

```
FirmaInfo.OstatniRok As Integer
```

Opis:

Identyfikator ostatniego założonego w firmie roku obrachunkowego.

Przykład

```
rokId = rok.OstatniRok
```

3.5.2.2 RokDenom

Składnia:

```
FirmaInfo.RokDenom As Integer
```

Opis:

Informacja o denominacji. Ustawione na wartość 1 tylko dla firm, które rozpoczęły używanie programu w roku 1994 i przechodziły przez denominację 1-01-1995 roku.

Przykład

```
rokId = rok.RokDenom
```

3.5.2.3 RokCount

Składnia:

```
FirmaInfo.RokCount As Integer
```

Opis:

Ilość lat obrachunkowych

Przykład

```
i = 0  
while i < database.FirmaInfo.RokCount  
    MsgBox database.FirmaInfo.Rok(i).Katalog  
    i = i + 1  
wend
```

3.5.2.4 Rok

Składnia:

```
FirmaInfo.Rok(rokId As Integer) As RokInfo
```

Opis:

Pobranie obiektu określonego roku obrachunkowego.

Przykład

```
MsgBox database.FirmaInfo.Rok(i).Katalog
```

3.5.2.5 CreateDate

Składnia:

```
FirmaInfo.CreateDate As Data
```

Opis:

Data utworzenia danych firmy.

Przykład

```
database.FirmaInfo.CreateDate
```

3.5.3 Pozostałe własności umożliwiające odczytanie nazwy, danych adresowych oraz identyfikacyjnych firmy.

- 3.5.3.1 Nazwa
- 3.5.3.2 NazwaSkrócona
- 3.5.3.3 Wojewodztwo
- 3.5.3.4 Miejscowosc
- 3.5.3.5 Ulica
- 3.5.3.6 Dom
- 3.5.3.7 Lokal
- 3.5.3.8 KodPocztowy
- 3.5.3.9 Poczta
- 3.5.3.10 NIP
- 3.5.3.11 Regon
- 3.5.3.12 Telefon

Przykład

```
MsgBox "Nazwa: " + database.FirmaInfo.Nazwa + Chr(10) + _  
      "Nazwa skrócona: " + database.FirmaInfo.NazwaSkrócona + Chr(10)+ _  
      "Adres: " + Chr(10) + _  
database.FirmaInfo.KodPocztowy + " " + _  
database.FirmaInfo.Poczta + Chr(10) + _  
      database.FirmaInfo.Ulica + " " + _  
database.FirmaInfo.Dom + "/" + database.FirmaInfo.Lokal
```

3.6 RokInfo

Obiekt RokInfo umożliwia uzyskanie informacji o latach obrachunkowych w programie Finanse i Księgowość.

3.6.1 Właściwości

- 3.6.1.1 RokId

Składnia:

```
RokInfo.RokId As String
```

Opis:

Identyfikator roku.

Przykład

```
rokId = rok.RokId
```

- 3.6.1.2 Katalog

Składnia:

```
RokInfo.Katalog As String
```

Opis:

Właściwość umożliwia pobranie katalogu roku.

Przykład

```
path = path + "\" + rok.Katalog
```

3.6.1.3 PoczatekRoku

Składnia:

```
RokInfo.PoczatekRoku As Data
```

Opis:

Właściwość umożliwia pobranie daty początku roku obrachunkowego.

Przykład

```
data = rok.PoczatekRoku
```

3.6.1.4 KoniecRoku

Składnia:

```
RokInfo.KoniecRoku As Data
```

Opis:

Właściwość umożliwia pobranie daty końca roku obrachunkowego.

Przykład

```
data = rok.KoniecRoku
```

3.6.1.5 DlugoscRoku

Składnia:

```
RokInfo.DlugoscRoku As Integer
```

Opis:

Właściwość umożliwia pobranie długości roku w miesiącach.

Przykład

```
months = rok.DlugoscRoku
```

3.6.1.6 Zamkniety

Składnia:

```
RokInfo.Zamkniety As Boolean
```

Opis:

Właściwość określająca czy rok jest zamknięty.

Przykład

```
bClosed = rok.Zamkniety
```

3.6.1.7 WArchiwum

Składnia:

```
RokInfo.WArchiwum As Boolean
```

Opis:

Czy rok został zarchiwizowany i jego dane usunięte z dysku.

Przykład

```
bArchiw = rok.WArchiwum
```

3.6.1.8 KsięgowaniaRownolegleAutomatyczne

Składnia:

```
RokInfo.KsięgowaniaRownolegleAutomatyczne As Boolean
```

Opis:

Czy stosowane są równoległe księgowana automatyczne.

Przykład

```
bKRA = rok.KsięgowaniaRownolegleAutomatyczne
```

3.6.1.9 Koszty

Składnia:

```
RokInfo.Koszty As String
```

Opis:

Właściwość określa, jaki typ rozliczania kosztów jest stosowany.

Przykład

```
typ = rok.Koszty
```

3.7 SyncroSubject

Obiekt SyncroSubject umożliwia synchronizację danych kartotekowych z danymi programu Finanse i Księgowość. Obecnie w zależności od opcji otwarcia potrafi synchronizować dane kontrahentów stałych i incydentalnych, pracowników oraz urzędów skarbowych, kartoteki krajów i walut. w zależności od wybranej kartoteki obiekt ten może korzystać z mechanizmu zwracania nowych lub zmienionych (dla kolejnych porównań) rekordów oraz funkcje prostego automatycznego uzgadniania danych.

3.7.1 Metody

3.7.1.1 Open

Składnia:

```
SyncroSubject.Open (pBtDatabase As BtDatabase, type as SubjectType)
```

Opis:

Metoda, którą powinniśmy rozpoczynać pracę z obiektem synchronizacji. Jako parametry powinna otrzymać otwarte połączenie do bazy danych oraz typ obiektów, z którymi chcemy dokonać uzgodnień. Mamy do dyspozycji typy obiektów:

SUB_KONTRAHENCI (2) – kontrahenci

SUB_KH_JEDNOKROTNI (8) – kontrahenci incydentalni

SUB_PRACOWNICY (1) – pracownicy

SUB_URZEDY (4) – urzędy

SUB_KRAJ (9) – kraje

SUB_WALUTY (5) – waluty

SUB_SLOWNIK (99) – słowniki

Przykład

```
syncro.Open database, SUB_URZEDY
```


3.7.1.2 GetRecById

Składnia:

```
SyncroSubject.GetRecById (nId As Long) As Integer
```

Opis:

Metoda dokonuje odczytu obiektu o identyfikatorze nId oraz typie podanym w metodzie Open. Jako wynik zwraca wartość 0 lub kod błędu. Dane przeczytanego rekordu są dostępne poprzez właściwość **Value**

Przykład

```
status = syncro.GetRecById(50)
```

3.7.1.3 GetRecByPoz

Składnia:

```
SyncroSubject.GetRecByPoz (nPoz As Long) As Integer
```

Opis:

Metoda dokonuje odczytu obiektu o wartości atrybutu „pozycja” podanej w nPoz oraz typie podanym w metodzie Open. Jako wynik zwraca wartość 0 lub kod błędu. Dane przeczytanego rekordu są dostępne poprzez właściwość **Value**. Atrybut „pozycja” jest dostępny tylko dla podmiotu: kontrahent, pracownik, urząd, oraz słowników i dla tych metoda ma zastosowanie. Są to podmioty używane, jako kartoteki dla planu kont i numer konta jest związany z atrybutem „pozycja”.

Przykład

```
status = syncro.GetRecByPoz(50) ' odczyt podmiotu o 'numerze' 50
```

3.7.1.4 GetFirstRec

Składnia:

```
SyncroSubject.GetFirstRec () As Integer
```

Opis:

Metoda dokonuje odczytu pierwszego rekordu danego typu. Dane przeczytanego rekordu są dostępne poprzez właściwość **Value**

Przykład

```
status = syncro.GetFirstRec
```

3.7.1.5 GetFirstNewRec

Składnia:

```
SyncroSubject.GetFirstNewRec () As Integer
```

Opis:

Metoda umożliwia dokonanie synchronizacji nowych i zmienionych rekordów. w tym celu należy najpierw zapamiętać własności LastModification, LastId i RecordCount, następnie ustawić te własności wartościami z poprzedniej synchronizacji i wywołać metodę, aby dostać kolekcję rekordów, dla których wymagana jest synchronizacja (dane zostały zmienione). Metoda ma zastosowanie dla typu kontrahent i pracownik.

Przykład

```
syncro.LastId = oldId  
syncro.LastModification = oldDate  
syncro.RecordCount = oldCount  
status = syncro.GetFirstNewRec  
while status = 0  
    status = syncro.GetNext  
wend
```

3.7.1.6 GetNext

Składnia:

```
SyncroSubject.GetNext () As Integer
```

Opis:

Pobiera następny rekord z kolekcji otwartej przez metody GetFirstNewRec oraz GetFirstRec

Przykład

```
status = syncro.GetFirstRec  
while status = 0  
    status = syncro.GetNext  
wend
```

3.7.1.7 ClearRecord

Składnia:

```
SyncroSubject.ClearRecord ()
```

Opis:

Metoda czyści zawartość rekordu. Należy ją wywołać przed ustawieniem wartości pól w celu wstawienia nowego rekordu.

Przykład

```
syncro.ClearRecord
```

3.7.1.8 Insert

Składnia:

```
SyncroSubject.Insert (nRecLen As Long) As Integer
```

Opis:

Wstawia nowy rekord a zwraca 0 lub kod błędu. Metoda nie jest implementowana dla typu kraj.

Przykład

```
status = syncro.Insert
```

3.7.1.9 Update

Składnia:

```
SyncroSubject.Update (nRecLen As Long) As Integer
```

Opis:

Uaktualnia bieżący rekord a zwraca 0 lub kod błędu.

Parametr nRecLen jest opcjonalny i powinien być podany tylko w przypadku rekordów danych o zmiennej długości.

UWAGA. Metoda nie jest implementowana dla typu kraj.

Przykład

```
status = syncro.Update
```

3.7.1.10 Validate

Składnia:

```
SyncroSubject.Validate () As Boolean
```

Opis:

Dokонуje wstępnej walidacji danych. Powinna zostać zawołana przed metodami Insert oraz Update.

Przykład

```
bValid = syncro.Validate
```

3.7.1.11 CheckIfExists

Składnia:

```
SyncroSubject.CheckIfExists() As Integer
```

Opis:

Wraz z metodą GetExistingId stanowi prosty automatyczny mechanizm uzgadniania danych w obrębie danego typu obiektu. w tym celu należy wypełnić obiekt wartościami a następnie wywołać metodę CheckIfExists, która zwraca ilość obiektów o najwyższym stopniu podobieństwa do wstawionych danych. Wywołanie następnie metody GetExistingId z indeksem z zakresu od, (0) do (count-1) umożliwi pobranie odpowiedniego id podobnego rekordu.

UWAGA. Metoda nie jest implementowana dla typu kraj.

Przykład

```
count = syncro.CheckIfExists  
i = 0  
while i < count  
    id = syncro.GetExistingId(i)  
    i = i + 1  
wend
```

3.7.1.12 GetExistingId

Składnia:

```
SyncroSubject.GetExistingId (index As Integer) As Long
```

Opis:

Opis zawarty przy opisie metody CheckIfExists.

Przykład

```
count = syncro.CheckIfExists  
i = 0  
while i < count  
    id = syncro.GetExistingId(i)  
    i = i + 1  
wend
```

3.7.2 Właściwości

3.7.2.1 Type

Składnia:

```
SyncroSubject.Type As SubjectType
```

Opis:

Zwraca typ obiektu synchronizacji.

Przykład

```
type = syncro.Type
```

3.7.2.2 LastModification

Składnia:

```
SyncroSubject.LastModification As SubjectType
```

Opis:

Właściwość umożliwiająca odczytanie, kiedy ostatnio była modyfikowana dana tabela oraz ustawienie parametru dla metody GetFirestNewRecord.

Przykład

```
lastDate = syncro.LastModification  
syncro.LastModification = oldDate
```

3.7.2.3 LastId

Składnia:

```
SyncroSubject.LastId As Long
```

Opis:

Właściwość umożliwiająca odczytanie maksymalnego identyfikatora rekordu oraz ustawienie parametru dla metody GetFirestNewRecord.

Przykład

```
lastId = syncro.LastId  
syncro.LastId = oldId
```

3.7.2.4 RecordCount

Składnia:

```
SyncroSubject.RecordCount As Long
```

Opis:

Właściwość umożliwiająca odczytanie ilości rekordów danego typu oraz ustawienie parametru dla metody GetFirestNewRecord.

Przykład

```
newCount = syncro.RecordCount  
syncro.RecordCount = oldCount
```

3.7.2.5 Value

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych.

UWAGA. Każdy z typów udostępnianych przez interfejs SyncroSubject posiada własną unikalną kolekcję atrybutów. Dla każdego typu można ją przejrzeć dokonując wywołania kodu:

Przykład:

```
For i = 0 to sync.FieldCount()-1  
    MsgBox Cstr(i) + "=" + sync.FieldName(i)  
Next i
```

3.7.2.6 FieldName

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

3.7.2.7 FieldType

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

3.7.2.8 FieldSize

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

3.7.2.9 FieldsCount

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

3.7.2.10 BtDatabase

Składnia:

```
SyncroSubject.BtDatabase As BtDatabase
```

Opis:

Właściwość umożliwia pobranie aktywnego połączenia, na którym pracuje obecnie obiekt synchronizacji.

Przykład

```
set database = syncro.BtDatabase
```

3.8 Dokument

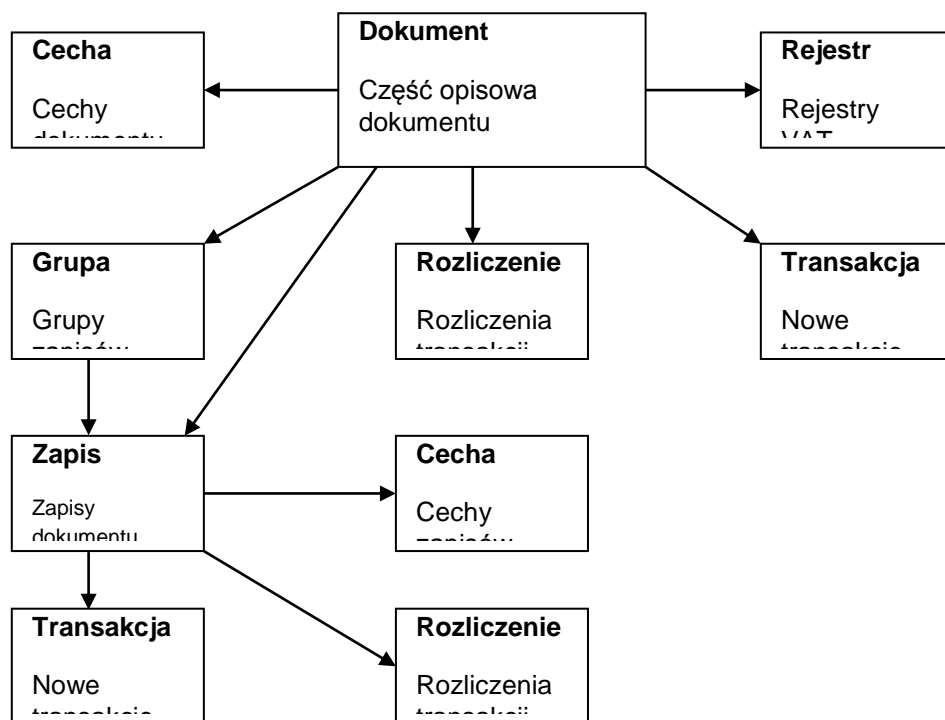
Obiekt Dokument reprezentuje część opisową dokumentu.

Nazwy pól danych, dostępnych poprzez ten obiekt oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "Dokumenty" w bazie danych programu. Poprzez ten obiekt realizowany jest dostęp do wszystkich informacji związanych z dokumentem: Zapisów księgowych, rejestrów VAT, rozrachunków oraz cech. Obiekt reprezentuje więc pełny dokument księgowy w programie Finanse i Księgowość.

3.8.1 Struktura obiektu

Poprzez obiekt **Dokument** realizowany jest dostęp do wszystkich informacji związanych z dokumentem: Części opisowej, zapisów księgowych, rejestrów VAT, rozrachunków oraz cech.

Dostęp do pól części opisowej dokumentu realizowany jest bezpośrednio poprzez pola danych obiektu **Dokument**, a dostęp do zapisów, rozrachunków i cech jest zorganizowany poprzez dodatkowe obiekt, będące właściwościami obiektu **dokument**. Struktura tych obiektów jest przedstawiona na rysunku.



3.8.2 Struktura obiektów związanych z dokumentem księgowym

Należy zwrócić uwagę, że niektóre obiekty występują na diagramie dwukrotnie, jednakże w innych miejscach hierarchii obiektów. Obiekt **Cecha** reprezentuje cechy dokumentów, oraz cechy zapisów. Podobnie nowe transakcje i rozliczenia mogą być związane albo z poszczególnymi zapisami księgowymi, albo z całym dokumentem. Szczegółowo zostało to wyjaśnione przy opisie obiektu **Transakcja**.

Zapisy dokumentu mogą być tworzone, jako elementy grupy zapisów, oraz jako elementy bezpośrednio zapisów. w przypadku używania grup zapisów, całość logiki związanej z tworzeniem bilansującej się grupy zapisów oraz tworzeniem rozbić i numeracją pozycji i rozbić wykonywana jest automatycznie, z tego względu zalecane jest używanie obiektu grupy. Jeśli zapisy tworzone są bezpośrednio przy obiekcie dokumentu, należy przestrzegać zasad opisanych przy obiekcie **Zapis**.

3.8.3 Właściwości

3.8.3.1 Value

Składnia:

```
Dokument.Value ( Index as VARIANT) As VARIANT
```

Opis:

Właściwość Value umożliwia pobranie lub ustawienie wartości dowolnego pola w części opisowej dokumentu.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "Dokumenty" w bazie danych programu. Lista ta jest jednak różna dla wersji Symfonia i Symfonia ERP.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Od wersji 2010 w wersji Symfonia ERP dostępne jest pole 'Uid' służące jednoznacznej identyfikacji dokumentu.

Przykład

```
Dok.Value("tresc") = "Usługa prawna 12/m/2009"  
Dok("tresc") = "Usługa prawna 12/m/2009"
```

3.8.3.2 Grupa

Składnia:

```
Dokument.Grupa As Grupa
```

Opis:

Właściwość Grupa umożliwia dostęp do obiektu reprezentującego elementarną grupę bilansujących się zapisów księgowych. Obiekt Dokument umożliwia w danej chwili dostęp tylko do jednej, bieżącej grupy zapisów. Odpowiednie metody obiektu Grupa umożliwiają poruszanie się po kolekcji wszystkich grup oraz zapisów każdej grupy danego dokumentu, tworzenia nowych grup i zapisów oraz dalej zapewniają dostęp do rozrachunków związanych z tymi zapisami.

Przykład:

```
Set Gr = Dok.Grupa  
Gr.Insert  
set Zap = gr.Zapis  
Zap.Value("kwota") = 1234.56
```

3.8.3.3 Zapis

Składnia:

```
Dokument.Zapis As Zapis
```

Opis:

Właściwość Zapis umożliwia dostęp do obiektu reprezentującego jeden zapis księgowy, wprowadzający jedną kwotę na jedno konto po jednej stronie. Obiekt Dokument umożliwia w danej chwili dostęp tylko do jednego, bieżącego zapisu księgowego. Odpowiednie metody obiektu Zapis umożliwiają poruszanie się po kolekcji wszystkich zapisów danego dokumentu, tworzenia nowych zapisów oraz dalej zapewniają dostęp do rozrachunków związanych z tymi zapisami.

Przykład:

```
Set zap = Dok.Zapis  
Zap.Insert  
Zap.Value("kwota") = 1234.56
```

3.8.3.4 RejestrVAT

RejestrVAT:

```
Dokument.RejestrVAT As RejestrVAT
```

Opis:

Właściwość RejestrVAT umożliwia dostęp do obiektu reprezentującego jeden rejestr VAT, związanego z jedną definicją rejestru, oraz z jednym rodzajem zakupu A,B,C lub D. Obiekt Dokument umożliwia w danej chwili dostęp tylko do jednego, bieżącego rejestru VAT. Odpowiednie metody obiektu RejestrVAT umożliwiają poruszanie się po kolekcji wszystkich rejestrów danego dokumentu, oraz dodawanie nowych rejestrów.

Obiekt RejestrVAT jest zalecanym sposobem obsługi rejestrów Vat na dokumencie.

Przykład:

```
Set Rej = Dok.RejestrVAT  
Rej.Insert  
Rej.Value("netto") = 1234.56  
Rej.Value("stawka") = 22.0  
Rej.Value("def") = 7
```

3.8.3.5 Rejestr

Rejestr:

`Dokument.Rejestr` As `Rejestr`

UWAGA! Właściwość jest utrzymywana dla kompatybilności z wersjami wcześniejszymi niż 5.10. Ponieważ nie posiada pełnej funkcjonalności należy zamiast niej używać właściwości RejestrVAT. Właściwość Rejestr będzie utrzymywana do wersji 2010 włącznie.

Opis:

Właściwość Rejestr umożliwia dostęp do obiektu reprezentującego jeden rejestr VAT, związany z jedną definicją rejestru i wszystkimi zdefiniowanymi stawkami, oraz z jednym rodzajem zakupu A,B lub C. Obiekt Dokument umożliwia w danej chwili dostęp tylko do jednego, bieżącego rejestru VAT. Odpowiednie metody obiektu Rejestr umożliwiają poruszanie się po kolekcji wszystkich rejestrów danego dokumentu, oraz dodawanie nowych rejestrów.

Przykład:

```
Set Rej = Dok.Rejestr
Rej.Insert
Rej.Value("netto0") = 1234.56
```

3.8.3.6 Cecha

Składnia:

`Dokument.Cecha` As `Cecha`

Opis:

Właściwość Cecha umożliwia dostęp do obiektu reprezentującego jedną cechę dokumentu. Obiekt Dokument umożliwia w danej chwili dostęp tylko do jednej, bieżącej cechy dokumentu. Odpowiednie metody obiektu Cecha umożliwiają poruszanie się po kolekcji wszystkich cech danego dokumentu, oraz dodawanie nowych cech.

Przykład:

```
Set c = Dok.Cecha
c.Insert
c.Value("id") = 56
```

3.8.3.7 Transakcja

Składnia:

`Dokument.Transakcja` As `Transakcja`

Opis:

Właściwość Transakcja umożliwia dostęp do obiektu reprezentującego nową transakcję (tylko jedną), związaną z rozrachunkiem na koncie kontrahenta na fakturach oraz rachunkach. Odpowiednie metody obiektu Transakcja umożliwiają odczytanie istniejącej transakcji oraz utworzenie nowej.

Pełny opis tej właściwości został zamieszczony w dalszej części opracowania, przy opisie obiektu Transakcja.

Przykład:

```
Set tr = Dok.Transakcja
Tr.Insert
tr.Value("termin") = "2001-05-28"
```


3.8.3.8 Rozliczenie

Składnia:

```
Dokument.Rozliczenie As Rozliczenie
```

Opis:

Właściwość **Rozliczenie** umożliwia dostęp do obiektu reprezentującego jedno rozliczenie, związane z rozrachunkiem na koncie kontrahenta na fakturach oraz rachunkach. Obiekt **Rozliczenie** umożliwia w danej chwili dostęp tylko do jednego, bieżącego rozliczenia. Odpowiednie metody obiektu **Rozliczenie** umożliwiają poruszanie się po liście wszystkich istniejących rozliczeń związanych z danym zapisem księgowym, oraz tworzenie nowych rozliczeń.

Pełny opis tej właściwości został zamieszczony w dalszej części opracowania, przy opisie obiektu **Rozliczenie**.

Przykład:

```
Set ro = Dok.Rozliczenie  
ro.Insert  
ro.Value("dSymbol") = "FV12/01/0001"
```

3.8.3.9 EOF

Składnia:

```
Dokument.EOF As Boolean
```

Opis:

Właściwość **EOF** informuje, czy dokument został prawidłowo wczytany. Właściwość ta jest ustawiana przez następujące metody:

- MoveFirst
- MoveNext

Jest to właściwość tylko do odczytu.

Przykład:

```
Dok.MoveFirst  
While Not Dok.EOF  
    'robimy coś interesującego z tym dokumentem  
    Dok.MoveNext  
Wend
```

3.8.4 Metody

3.8.4.1 FieldName

Składnia:

```
Function Obiekt.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.8.4.2 FieldSize

Składnia:

```
Function Obiekt.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.8.4.3 FieldType

Składnia:

```
Function Obiekt.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.8.4.4 FieldDec

Składnia:

```
Function Obiekt.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.8.4.5 FieldCount

Składnia:

```
Function Obiekt.FieldCount( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.8.4.6 Open

Składnia:

```
Function Dokument.Open( Sel As String, Firma As BtDatabase ) As Boolean
```

Parametry:

Sel miejsce położenia dokumentów, opisane w następujący sposób:

“indeks_roku; położenie”

indeks_roku: 0,1...n - numeryczny indeks roku, pominięcie tego parametru oznacza ostatni rok działalności

Położenie:

BUFOR – bufor, do odczytu i zapisu

KSIĘGI – księgi, tylko do odczytu

WZORCE – wzorce, do odczytu i zapisu

Pominięcie tego parametru oznacza bufor

Firma Obiekt otwartego połączenia do bazy danych programu Finanse i Księgowość

Opis:

Metoda otwiera połączenie obiektu z buforem, księgami lub wzorcami programu Finanse i Księgowość.

Obiekt umożliwia przeglądanie wszystkich dokumentów wprowadzonych do programu FK, oraz wprowadzanie nowych dokumentów do wzorców lub bufora programu FK.

Przykład:

```
Dim dbs As New BtDatabase  
Dim src As New Pdokument
```

```
dbs.Open "c:\symfonia\mojafirma", "admin", "cqs2sp7"
src.Open "0;BUFOR", dbs 'bufor, pierwszy rok działalności
'odczytujemy lub zapisujemy dokumenty
```

3.8.4.7 MoveFirst

Składnia:

```
Function Dokument.MoveFirst() As Boolean
```

Opis:

Metoda powoduje wczytanie pierwszego dokumentu z bufora, wzorców lub ksiąg, według kolejności wprowadzania (księgowania) dokumentów.

Przykład:

```
If Dok.MoveFirst then
    MsgBox Dok.Value("id")
endif
```

3.8.4.8 MoveNext

Składnia:

```
Function Dokument.MoveNext() As Boolean
```

Opis:

Metoda powoduje wczytanie kolejnego dokumentu z bufora, wzorców lub ksiąg, według kolejności wprowadzania (księgowania) dokumentów.

Przykład:

```
Dok.MoveFirst
While not dok.EOF then
    MsgBox Dok.Value("id")
    Dok.MoveNext
Wend
```

3.8.4.9 Insert

Składnia:

```
Function Dokument.Insert() As Boolean
```

Opis:

Metoda dodaje nowy dokument do bufora lub wzorców programu FK. Przed dodaniem nowego dokumentu przeprowadzana jest kontrola poprawności dokumentu, a wszystkie błędy, komunikaty oraz informacje z tej kontroli dostępne są poprzez obiekt **DocErrors**.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli dokument został zapisany do bufora, jeśli dokument nie zostanie zapisany (z powodu błędów uniemożliwiających jego wprowadzenie lub błędów obsługi bazy danych) metoda zwraca wartość FALSE

Przykład:

```
If Not Dok.Insert then
    MsgBox "dokument nie został zapisany"
Endif
```

3.8.4.10 Delete

Składnia:

```
Function Dokument.Delete() As Boolean
```

Opis:

Metoda kasuje bieżący, wczytany dokument w buforze lub wzorcach.

Kasowany jest cały, kompletny dokument, czyli nagłówek, zapisy, cechy oraz rozrachunki. Przed skasowaniem dokument musi być wczytany jednym z poleceń: **MoveFirst**, **MoveNext**, **GetByID** lub **GetByNr**.

Polecenie nie umożliwia skasowania dokumentu zaksięgowanego.

Dokument może nie zostać skasowany, jeśli transakcja utworzona przez ten dokument jest rozliczona przez inny dokument. Taki dokument może być usunięty tylko po ręcznym usunięciu rozliczeń nowych transakcji tego dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli dokument został skasowany.

Jeśli dokument nie zostanie skasowany metoda zwraca wartość FALSE

Przykład:

```
If Dok.GetbyNr("DP", 1) then
    Dok.Delete
Endif
```

3.8.4.11 Update

Składnia:

```
Function Dokument.Update() As Boolean
```

Opis:

Metoda niezaimplementowana. w przyszłych wersjach umożliwi zapisanie zmian dokonanych w dokumencie. Przed zapisaniem zmian dokument musi być wczytany jednym z poleceń: MoveFirst, MoveNext, GetByID lub GetByNr. Przed zapisaniem przeprowadzana jest kontrola poprawności dokumentu, a wszystkie błędy, komunikaty oraz informacje z tej kontroli dostępne są poprzez obiekt DocErrors.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli dokument został zapisany do bufora, jeśli dokument nie zostanie zapisany (z powodu błędów uniemożliwiających jego wprowadzenie lub błędów obsługi bazy danych) metoda zwraca wartość FALSE

3.8.4.12 Validate

Składnia:

```
Function Dokument.Validate() As Boolean
```

Opis:

Metoda sprawdza poprawność dokumentu, oraz dokonuje automatycznych korekt i uzupełniania danych.

W szczególności np. dla dokumentów typu faktura dokonuje automatycznych poprawek, co do zgodności danych w obiekcie rejestru z danymi w dokumencie: NIP, kontrahent, numer własny, numer ewidencyjny.

Wszystkie błędy, komunikaty oraz informacje z tej kontroli dostępne są poprzez obiekt DocErrors.

Przykład:

```
Dim err as new DocErrors
'...
Dok("skrot") = "DP"
Dok("Numer") = 1
Dok.Zapis.Insert
Dok.Zapis("Kwota") = 1234
Dok.Zapis("Strona") = 0
```

```
Dok.Zapis("Kwota") = 1230
Dok.Zapis("Strona") = 1
Dok.Validate
Msgbox "Ilość komunikatów: " + Cstr(err.Count)
```

3.8.4.13 Clear

Składnia:

```
Sub Dokument.Clear()
```

Opis:

Metoda zeruje wszystkie pola dokumentu, usuwa wszystkie zapisy, cechy, rejestry VAT oraz rozrachunki.

Metodę tę należy przed rozpoczęciem wprowadzania danych dokumentu, w szczególności po zapisaniu jednego dokumentu, a przed rozpoczęciem wprowadzania kolejnego dokumentu

Przykład:

```
Dok.Clear
'wypełnienie danymi dokumentu 1
Dok.Insert
Dok.Clear
'wypełnienie danymi dokumentu 2
Dok.Insert
```

3.8.4.14 GetByID

Składnia:

```
Function Dokument.GetByID( Id As long) As Boolean
```

Opis:

Metoda umożliwia wczytanie dokumentu o określonym identyfikatorze (id). Metoda zwraca wartość TRUE, jeśli dokument istnieje oraz został prawidłowo wczytany.

Przykład:

```
If Dok.GetbyID(1)
    MsgBox Dok("tresc")
endif
```

3.8.4.15 GetByNR

Składnia:

```
Function Dokument.GetByNR( Skrot As String, Numer As Long) As Boolean
```

Opis:

Metoda umożliwia wczytanie dokumentu o określonym numerze ewidencyjnym (skrót + numer w ramach skrótu). Metoda zwraca wartość TRUE, jeśli dokument istnieje oraz został prawidłowo wczytany.

Przykład:

```
If Dok.GetbyNr("DP", 1)
    MsgBox Dok("tresc")
Endif
```

3.8.4.16 GetByGuid

Składnia:

```
Function Dokument.GetByGuid( guid As BSTR) As Boolean
```

Opis:

Metoda umożliwia wczytanie dokumentu o określonym globalnym identyfikatorze (guid). Metoda zwraca wartość TRUE, jeśli dokument istnieje oraz został prawidłowo wczytany.

Przykład:

```
If Dok.GetbyGuid("3F708037-ADA1-0007-A4C8-7270644020F7")
    MsgBox Dok("tresc") + Dok("nazwa")
endif
```

3.8.5 Zdarzenia

Obiekt nie generuje zdarzeń.

3.9 Zapis

3.9.1 Opis

Obiekt reprezentuje księgowanie jednej kwoty na jednym koncie po jednej jego stronie. z tym obiektem związane są dodatkowo obiekty cech, transakcji oraz rozliczeń związanych z tym zapisem księgowym.

Obiekt zawiera metody umożliwiające dostęp do wszystkich zapisów wczytanego dokumentu. Jest to dostęp sekwencyjny, co oznacza, że w danej chwili możliwy jest dostęp do jednego i tylko jednego, bieżącego zapisu dokumentu. Obiekt umożliwia edycję, nawigowanie, dodawanie oraz usuwanie zapisów dokumentu.

3.9.2 Zasady tworzenia wierszy w tabeli zapisów

Zapisy dokumentu mogą być tworzone, jako elementy grupy zapisów, oraz jako elementy bezpośrednio zapisów. w przypadku używania grup zapisów, całość logiki związanej z tworzeniem bilansującej się grupy zapisów oraz tworzeniem rozbić i numeracją pozycji i rozbić wykonywana jest automatycznie, z tego względu zalecane jest używanie obiektu grupy. Jeśli zapisy tworzone są bezpośrednio przy obiekcie dokumentu, należy przestrzegać zasad przedstawionych poniżej.

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 rozbitcia 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).

Tabeli zapisów umieszczane są kolejne wiersze zapisów (minimum dwie). Numery pozycji i rozbicia są polami tabeli zapisu. Prawidłowe numery pozycji i rozbicia muszą być nadane zgodnie z podanymi wyżej zasadami.

UWAGA! Jeśli będziemy korzystać z obiektu grupa, wówczas nie musimy jawnie specyfikować wartości: pozycja oraz rozbicie patrz Grupa

Przykłady numeracji pozycji i rozbicia dla różnych schematów księgowania i odpowiadające im wiersze w tabeli zapisów w dokumencie.

P - oznacza numer pozycji

R - oznacza numer rozbicia

3.9.2.1 Pojedyncza pozycja w dokumencie

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

Kolejne pozycje tworzone w obiekcie **zapis**

Strona	Kwota	Konto	Pozycja	Rozbicie
0	345,23	100	0	0
1	345,23	201-1	0	1

3.9.2.2 Pojedyncza pozycja w dokumencie 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

Kolejne pozycje tworzone w obiekcie **zapis**

Strona	Kwota	Konto	Pozycja	Rozbicie
0	345,23	100	0	0
1	200,00	010-1	0	1
1	40,00	010-2	0	2
1	105,23	010-3	0	3

3.9.2.3 Pojedyncza pozycja w dokumencie 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				

Kolejne pozycje tworzone w obiekcie **zapis**

Strona	Kwota	Konto	Pozycja	Rozbicie
0	200	010-1	0	0
1	345,23	100	0	1
0	40,00	010-2	0	2
0	105,23	010-3	0	3

3.9.2.4 Trzy pozycje różnych typów na jednym dokumencie

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				

Kolejne pozycje tworzone w obiekcie **zapis**

Strona	Kwota	Konto	Pozycja	Rozbicie
0	345,23	100	0	0
1	200	010-1	0	1
1	143,23	010-2	0	2
0	84,12	100	1	0
1	84,12	201-1	1	1
0	3800,00	010-1	2	0
1	3865,00	100	2	1
0	65,00	010-2	2	2

3.9.3 Księgowania równoległe

Istnieją dwa typy księgowañ równoległych: na życzenie, oraz automatyczne księgowania równoległe dla kont zespołu "4" (kont kosztowych). Księgowania równoległe na życzenie mogą być tworzone w każdym wypadku, księgowania automatyczne tylko, jeśli firma prowadzi rachunek kosztów z kontami zespołu "4", "5" oraz kontem 490

Aby prawidłowo oznaczyć księgowania równoległe i powiązać je z odpowiednimi księgowaniami podstawowymi, w tabeli zapisów występuje kolumna o nazwie *księgowaniaRównoległe* oznaczone symbolicznie **Kr**.

Dzięki odpowiedniemu powiązaniu zapisów program kontroluje m.in. zgodność kwot księgowień podstawowych i równoległych, aktualizuje kwoty księgowień równoległych po zmianie księgowień podstawowych itd.

Wartości znaczników podane są w zapisie szesnastkowym. Część wartości może być łączona bitowo (suma logiczna), a część z nich nawzajem się wyklucza.

Wartości, które mogą znajdować się w tej kolumnie są następujące (w nawiasach wartości dziesiętne):

0x01 (1)	Ten wiersz jest zapisem równoległym
0x02 (2)	Dla tego wiersza utworzono zapis równoległy na życzenie
0x04 (4)	Dla tego wiersza utworzono zapis równoległy automatyczny
0x08 (8)	Tylko dla raportu kasowego: To jest wiersz księgowania równoległego automatycznego po stronie WN
0x10 (16)	Tylko dla raportu kasowego: To jest wiersz księgowania równoległego automatycznego po stronie MA
0x20 (32)	Ten wiersz jest zapisem równoległym na życzenie
0x40 (64)	To jest zapis równoległy, który występuje bezpośrednio po zapisie podstawowym, dla którego został utworzony i nie wchodzi do księgowania równoległego zbiorczego

T – połączenie dozwolone,

N – połączenie zabronione

? – połączenie nie zabronione, ale nie stosowane

	0x01	0x02	0x04	0x08	0x10	0x20	0x40
0x01	-	N	N	T	T	T	T
0x02	N	-	?	N	N	N	N
0x04	N	?	-	N	N	N	N
0x08	T	N	N	-	N	N	N
0x10	T	N	N	N	-	N	N
0x20	T	N	N	N	N	-	?
0x40	T	N	N	N	N	?	-

Znaczniki 0x02 i 0x04 mogą być użyte **wyłącznie** w zapisie podstawowym, pozostałe **wyłącznie** w zapisach równoległych.

3.9.4 Księgowania równoległe automatyczne

Księgowania równoległe automatyczne są tworzone automatycznie dla każdego zapisu na konto zespołu "4" (oczywiście oprócz konta rozliczenia kosztów zespołu "4" czyli konta 490).

UWAGA: w programie nie ma księgowień równoległych automatycznych z kont zespołu "5" na konta zespołu "4"

Księgowania równoległe automatyczne mogą być tworzone są na dwa sposoby, jako jeden lub dwa zbiorcze dekrety występujące po wszystkich zapisach podstawowych (na końcu dokumentu), lub jako księgowania bezpośrednio występujące po księgowaniach podstawowych na konta "4".

Pierwszy sposób (dekrety zbiorcze) jest historycznie pierwszy, oraz prostszy w realizacji.

Drugi sposób, księgowania tzw "przyklejone" do dekretów podstawowych powstały dopiero w wersji 3.0 programu Finanse i Księgowość.

3.9.5 Księgowania równoległe zbiorcze

Dla tego typu księgowania tworzone są zbiorcze dekrety na konto 490, sumujące kwoty wszystkich kont zespołu "4" po każdej stronie, a następnie dowolnie rozksięgowane na konta zespołu "5". Jeśli w księgowaniach podstawowych występują konta zespołu "4" tylko po jednej stronie, wówczas tworzony jest jeden zbiorczy dekret, w przeciwnym wypadku tworzone są dwa dekrety, oddzielnie dla sumy po stronie WN i oddzielnie dla sumy po stronie MA

3.9.6 Księgowania równoległe zbiorcze automatyczne

Strona WN					Strona MA				
P	R	Kr	Kwota	Konto	P	R	Kr	Kwota	Konto
0	0	0x40	345,23	401	0	1	0	200,00	010-1
					0	2	0	143,32	010-2
1	0	0x40	84,12	402	1	1	0	84,12	201-1
2	0	0	200	010	2	1	0	200	100

Księgowanie równoległe

3	0	0x01	300,00	501	3	1	0x01	429,35	490
3	2	0x01	129,35	502					

W przypadku księgowania zbiorczych automatycznych wykorzystywane są wyłącznie znaczniki

0x40 dla księgowania podstawowych na konta zespołu "4" oraz 0x01 dla księgowania równoległych

3.9.7 Właściwości

3.9.7.1 Cecha

Składnia:

```
Zapis.Cecha As Cecha
```

Opis:

Właściwość Cecha umożliwia dostęp do obiektu reprezentującego jedną cechę zapisu. Obiekt Cecha umożliwia w danej chwili dostęp tylko do jednej, bieżącej cechy dokumentu. Odpowiednie metody obiektu Cecha umożliwiają poruszanie się po kolekcji wszystkich cech danego zapisu, oraz dodawanie nowych cech.

Przykład:

```
Set c = Zap.Cecha
c.Insert
c.Value("id") = 56
```

3.9.7.2 Transakcja

Składnia:

```
Zapis.Transakcja As Transakcja
```

Opis:

Właściwość Transakcja umożliwia dostęp do obiektu reprezentującego nową transakcję (tylko jedną), związaną z rozrachunkiem na danym koncie. Odpowiednie metody obiektu Transakcja umożliwiają odczytanie istniejącej transakcji oraz utworzenie nowej.

Przykład:

```
Set tr = Zap.Transakcja
Tr.Insert
tr.Value("termin") = "2001-05-28"
```

3.9.7.3 Rozliczenie

Składnia:

```
Zapis.Rozliczenie As Rozliczenie
```

Opis:

Właściwość Rozliczenie umożliwia dostęp do obiektu reprezentującego jedno rozliczenie, związane z danym zapisem na danym koncie. Obiekt Rozliczenie umożliwia w danej chwili dostęp tylko do jednego, bieżącego rozliczenia. Odpowiednie metody obiektu Rozliczenie umożliwiają poruszanie się po liście istniejących rozliczeń oraz tworzenie nowych.

Przykład:

```
Set ro = Zap.Rozliczenie
ro.Insert
ro.Value("dSymbol") = "FV12/01/0001"
```

3.9.7.4 EOF

Składnia:

```
Zapis.EOF As Boolean
```

Opis:

Właściwość **EOF** informuje, czy dany zapis istnieje i został ustawiony, jako bieżący. Właściwość ta jest ustawiana przez następujące metody:

- MoveFirst
- MoveNext

Jest to właściwość tylko do odczytu.

Przykład:

```
Zap.MoveFirst
While Not Zap.EOF
    'robimy coś interesującego z tym zapisem
    Zap.MoveNext
Wend
```

3.9.7.5 Konto

Składnia:

```
Zapis.Konto As String
```

Opis:

Właściwość **Konto** zawiera tekstową reprezentację pełnego numeru konta. Numer konta ma następującą postać:

"Synt-poz1-poz2-poz3-poz4-poz5"

gdzie:

- Synt** trzynastkowy numer syntetyki konta, odpowiada numerycznemu polu "synt"
- Poz1** numer pierwszego poziomu analityki, o ile występuje, odpowiada numerycznemu polu "poz1"
- Poz2** numer drugiego poziomu analityki, o ile występuje, odpowiada numerycznemu polu "poz2"
- Poz3** numer trzeciego poziomu analityki, o ile występuje, odpowiada numerycznemu polu "poz3"
- Poz4** numer czwartego poziomu analityki, o ile występuje, odpowiada numerycznemu polu "poz4"
- Poz5** numer piątego poziomu analityki, o ile występuje, odpowiada numerycznemu polu "poz5"

Przykład:

```
Dim z as Zapis
Set z = dok.zapis
'użycie pola "konto"
z.Insert
z.value("konto") = "501-1-12"
'i równowazne uzycie pól numerycznych
z.insert
z.value("synt") = 501
z.value("poz1") = 1
z.value("poz2") = 12
```

3.9.7.6 Parent

Składnia:

```
Zapis.Parent As Object
```

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **Dokument**, do którego należy dany zapis.

Przykład:

```
Dim z as zapis
Dim d as Dokument
...
set d = z.parent
d.Value("datadok") = "2001-07-10"
```

3.9.7.7 Value

Składnia:

```
Zapis.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w zapisie.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają zawartości tabeli "Zapisy" w bazie danych programu i mogą być nieco różne w linii Symfonia i Symfonia ERP.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Od wersji 2010 w wersji Symfonia ERP dostępne jest pole 'Uid' służące jednoznacznej identyfikacji zapisu.

Przykład

```
zap.Value("Kwota") = 123.45
```

```
zap("konto") = "201-1-12"
```

3.9.8 Metody

3.9.8.1 FieldName

Składnia:

```
Function Zapis.FieldName( Index As Integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.9.8.2 FieldSize

Składnia:

```
Function Zapis.FieldSize( Index As Variant) As Integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.9.8.3 FieldType

Składnia:

```
Function Zapis.FieldType( Index As Variant) As Integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.9.8.4 FieldDec

Składnia:

```
Function Zapis.FieldDec( Index As Variant) As Integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.9.8.5 FieldCount

Składnia:

```
Function Zapis.FieldCount( Index As Variant) As Integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.9.8.6 MoveFirst

Składnia:

```
Function Zapis.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego pierwszego zapisu w grupie lub w dokumencie.

Przykład:

```
If zap.MoveFirst then  
    MsgBox "Kwota pierwszego zapisu=" + Cstr(zap.Value("kwota"))  
endif
```

3.9.8.7 MoveNext

Składnia:

```
Function Zapis.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego kolejnego zapisu w grupie lub dokumencie.

Przykład:

```
zap.MoveFirst
While not zap.EOF then
    MsgBox "Kwota zapisu=" + Cstr(zap.Value("kwota"))
    Dok.MoveNext
Wend
```

3.9.8.8 Insert

Składnia:

```
Function Zapis.Insert() As Boolean
```

Opis:

Metoda dodaje nowy zapis do grupy zapisów lub do dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli został utworzony kolejny zapis w grupie lub dokumencie.

Przykład:

```
If Not zap.Insert then
    MsgBox "nowy zapis nie może być utworzony"
Endif
```

3.9.8.9 Delete

Składnia:

```
Function Zapis.Delete() As Boolean
```

Opis:

Metoda usuwa bieżący zapis w grupie zapisów lub na dokumencie.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli zapis został usunięty.

Przykład:

```
If not zap.Delete() then
    MsgBox "zapis nie może być usunięty"
Endif
```

3.9.9 Zdarzenia

Obiekt nie generuje zdarzeń

3.10 Grupa

3.10.1 Opis

Obiekt **Grupa** reprezentuje grupę bilansujących się zapisów księgowych. Grupa składa się co najmniej z dwóch zapisów księgowych po dwóch przeciwnych stronach kont. Grupa może zawierać większą liczbę

zapisów księgowych, w takim przypadku po jednej stronie konta występuje dokładnie jeden zapis księgowy, a po drugiej stronie wszystkie pozostałe zapisy z grupy. Zapisy grupy zawsze muszą się bilansować.

Pierwszym zapisem w grupie jest zawsze zapis po stronie WN, następnym zawsze zapis po stronie MA. Wszystkie kolejne zapisy grupy (o ile występują) muszą być po tej samej stronie (wszystkie WN lub wszystkie MA) tworząc kolejne rozbicia kwoty na koncie. Koniecznie też w takim porządku należy dokładać elementy do kolekcji. Ilustrują to następujące tabele:

3.10.1.1 Pojedyncza pozycja w dokumencie z rozbiciem kwoty po stronie Wn

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

Kolejne zapisy tworzone w grupie

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

3.10.1.2 Pojedyncza pozycja w dokumencie z rozbiciem kwoty po stronie Ma

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

Kolejne zapisy tworzone w grupie

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

UWAGA! Obiekt Grupa automatycznie generuje strony kolejnych zapisów, oraz zapewnia prawidłową numerację pól *pozycja* oraz *rozbicie*. Oznacza to ni mniej ni więcej, że używając grup można „nie pamiętać” o nadawaniu wartości polom :strona, pozycja oraz rozbicie.

W jednym dokumencie może być utworzonych dowolnie dużo grup zapisów, każda z nich rozbita po dowolnej stronie. Również księgowania równoległe muszą być utworzone jako oddzielne grupy zapisów.

Przykład:

Dim dok as new Dokument

```

Dim g as Grupa
Dim z as Zapis
'... inicjalizacja dokumentu
set g = dok.grupa
g.Insert    'tworzymy grupę
' nie specyfikując rozbicia grupy pozostawiamy default (rozbicie=-1),
' który oznacza brak rozbicia, czyli dopuścimy jeden zapis po winien i ' jeden
po ma w tej grupie
set z = g.Zapis
' bo pierwszy zapis jest tworzony automatycznie przy tworzeniu grupy
z.Value("kwota") = 345.23
z.Value("konto") = "100"
z.Insert    'kolejny zapis w grupie
z.Value("kwota") = 200.00
z.Value("konto") = "010-1"
' jeśli jednak chcemy zrobić rozbicie to:
g.rozbicie = 1 'rozbicia po stronie MA
z.Insert    'kolejny zapis (rozbicie) w grupie
z.Value("kwota") = 145.23
z.Value("konto") = "010-2"
'tu może być dodana kolejna grupa zapisów
g.Insert
'... dodajemy zapisy do nowej grupy
z.Insert
'... kolejne zapisy do grupy
dok.Insert  'zapisujemy dokument

```

3.10.2 Właściwości

3.10.2.1 Parent

Składnia:

`Grupa.Parent` As Object

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **Dokument**, do którego należy dana grupa zapisów.

Przykład:

```

Dim g as grupa
Dim d as Dokument
...
set d = g.parent
d.Value("datadok") = "2001-07-10"

```

3.10.2.2 Rownolegle

Składnia:

`Grupa.Rownolegle` As integer

Opis:

Właściwość określa, czy w danej grupie znajdują się zapisy księgowych równoległych.

Wartości (w nawiasach wartości dziesiętne), które może przyjmować to pole są następujące:

0x00 (0)	Grupa księgowości podstawowych
0x01 (1)	Grupa księgowości równoległych
0x08 (8)	Tylko dla raportu kasowego: grupa księgowania równoległego automatycznego po stronie WN
0x10 (16)	Tylko dla raportu kasowego: grupa księgowania równoległego automatycznego po stronie MA
0x20 (32)	Grupa księgowości równoległych na życzenie
0x40 (64)	Grupa księgowości równoległych, która występuje bezpośrednio po grupie zapisów podstawowych, dla której został utworzony i nie wchodzi do grupy księgowości równoległych zbiorczych

Mapa dozwolonych połączeń wartości bitowych pola **Rownolegle**:

T – połączenie dozwolone,

N – połączenie zabronione

? – połączenie nie zabronione, ale nie stosowane

	0x01	0x08	0x10	0x20	0x40
0x01	-	T	T	T	T
0x08	T	-	N	N	N
0x10	T	N	-	N	N
0x20	T	N	N	-	?
0x40	T	N	N	?	-

Przykład:

```
set g = dok.grupa
g.Insert    `tworzymy grupe
g.równolegle = 0 `grupa księgowości podstawowych
`...
g.Insert    `tworzymy grupe
g.równolegle = 1 + 32 `grupa księgowości równoległych na życzenie
`...
```

3.10.2.3 Rozbicie

Składnia:

```
Grupa.Rozbicie As integer
```

Opis:

Właściwość określa, czy w danej grupie występuje rozbicie kwot po jednej ze stron, a jeśli tak to, po której stronie. Zgodnie z tym co napisano wyżej ustawienie właściwości można pominąć.

Dozwolone wartości są następujące:

-1: nie ma rozbić, w grupie występują tylko dwa zapisy, pierwszy po stronie WN, drugi po stronie MA, na taką samą kwotę. Jest to wartość domyślna.

0 : w grupie występują rozbita po stronie WN. Pierwszy zapis jest po stronie WN, drugi po stronie MA, a kolejne zapisy znów po stronie WN. Suma kwot zapisów po stronie WN musi się równać kwocie zapisu po stronie MA

1 : w grupie występują rozbieżności po stronie MA. Pierwszy zapis jest po stronie WN, drugi i wszystkie następne po stronie MA. Kwota zapisu po stronie WN musi się równać sumie kwot zapisów po stronie MA.

Przykład:

```
Dim g as grupa
Dim z as Zapis
Set g = dok.Grupa
g.Insert      `tworzy też pierwszy zapis po stronie WN
g.rozbieżność = 1 `rozbieżności będą tylko po stronie MA
Set z = g.Zapis
z.Value("kwota") = 100.00
z.Insert      `zapis po stronie MA
z.Value("kwota") = 80.00
z.Insert      `kolejny zapis po stronie MA
z.Value("kwota") = 20.00
```

3.10.2.4 Zapis

Składnia:

```
Grupa.Zapis As Zapis
```

Opis:

Właściwość udostępnia obiekt reprezentujący jeden (bieżący) zapis w grupie zapisów. Poprzez obiekt grupy możliwy jest dostęp tylko i wyłącznie do zapisów tej jednej (bieżącej) grupy. Dostęp do wszystkich zapisów dokumentu jest możliwy poprzez właściwość dokumentu: **Dokument.Zapis**.

Przykład:

```
Dim g as grupa
Dim z as Zapis
Set g = dok.Grupa
g.Insert
Set z = g.Zapis
z.Value("kwota") = 234.56
```

3.10.3 Metody

3.10.3.1 Insert

Składnia:

```
Function Grupa.Insert() As Boolean
```

Opis:

Metoda **Insert** dodaje do dokumentu nową grupę bilansujących się zapisów. Razem z nową grupą tworzony jest automatycznie pierwszy zapis grupy (po stronie WN).

Tworzenie nowej grupy nie jest możliwe, jeśli w grupie nie ma co najmniej dwóch zapisów. w takim przypadku metoda **Insert** zwraca wartość **False**.

Przykład:

```
Dim dok as new Dokument
Dim g as Grupa
Dim z as Zapis
`... inicjalizacja dokumentu
set g = dok.grupa
g.Insert      `tworzymy grupę
set z = g.Zapis
```

```
`pierwszy zapis (WN) jest tworzony automatycznie przy tworzeniu grupy
z.Value("kwota") = 345.23
z.Value("konto") = "100"
z.Insert `kolejny zapis w grupie (MA)
z.Value("kwota") = 345.23
z.Value("konto") = "010-1"
```

3.10.4 Zdarzenia

Obiekt nie generuje zdarzeń.

3.11 Cecha

3.11.1 Opis

Obiekt reprezentuje cechę (bieżącą) nadaną dokumentowi lub zapisowi. Dostęp do wszystkich cech dokumentu lub zapisu jest sekwencyjny, przy użyciu metod **MoveFirst** oraz **MoveNext**.

3.11.2 Właściwości

3.11.2.1 Value

Składnia:

```
Cecha.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola cechy.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Przykład

```
cecha.Value("id") = 512
```

3.11.3 Metody

3.11.3.1 FieldName

Składnia:

```
Function Cecha.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.11.3.2 FieldSize

Składnia:

```
Function Cecha.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.11.3.3 FieldType

Składnia:

```
Function Cecha.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.11.3.4 FieldDec

Składnia:

```
Function Cecha.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.11.3.5 FieldCount

Składnia:

```
Function Cecha.FieldCount( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.11.3.6 MoveFirst

Składnia:

```
Function Cecha.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej pierwszej cechy zapisu lub dokumentu.

Przykład:

```
If cecha.MoveFirst then
    MsgBox "nazwa=" + Cstr(cecha.Value("nazwa"))
endif
```

3.11.3.7 MoveNext

Składnia:

```
Function Cecha.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej kolejnej cechy zapisu lub dokumentu.

Przykład:

```
cecha.MoveFirst
do loop
    MsgBox "id=" + Cstr(cecha.Value("id"))
    cecha.MoveNext
Wend
```

3.11.3.8 Insert

Składnia:

```
Function cecha.Insert() As Boolean
```

Opis:

Metoda dodaje nową cechę do zapisu lub do dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli została utworzona kolejna cecha.

Przykład:

```
If Not dok.cecha.Insert() then
    MsgBox "nowa cecha nie może być utworzona"
Endif
```

3.11.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.12 Transakcja

3.12.1 Opis

Obiekt **Transakcja** reprezentuje nową transakcję utworzoną dla pojedynczego zapisu księgowego lub dla całego dokumentu.

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 całym dokumentem (sumą księgowania na konto kontrahenta ze wszystkich zapisów), a nie z poszczególnymi zapisami na konta

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 pojedynczymi zapisami na kontach rozrachunkowych

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 podpiętą na ostatnim poziomie analityki kartotekę kontrahentów, pracowników lub urzędów.

Dla jednego dokumentu dla faktur i RU oraz dla jednego zapisu dla pozostałych typów dokumentów może być utworzona najwyżej jedna nowa transakcja, odpowiadająca znacznikowi *[X] Nowa transakcja* na formacie wprowadzania rozrachunków w programie FK, oraz dowolna liczba rozliczeń istniejących transakcji.

W przypadku łączenia nowej transakcji z rozliczeniami transakcji dla jednego zapisu, kwota tego zapisu musi równać się sumie kwoty nowej transakcji oraz wszystkich rozliczeń.

3.12.2 Właściwości

3.12.2.1 Value

Składnia:

```
Transakcja.Value ( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w transakcji.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Dostępne pola obiektu **Transakcja**:

- "kwota" Kwota nowej transakcji
- "kwota_w" kwota transakcji w walucie – tylko dla kont i dokumentów walutowych
- "stopa" wysokość stałej, dziennej stopy odsetek
- "ustawowe" znacznik odsetek ustawowych
- "termin" data terminu płatności
- "zaliczka" znacznik, że nowa transakcja jest zaliczką

- "rachBenef" tylko dla wersji Symfonia ERP, rachunek, na który ma być przelana zapłata

Dla transakcji walutowych wartość pola "kwota" musi być zgodna z wartością pola "kwota_w" przemnożoną przez kurs waluty w zapisie lub dokumencie.

Przykład

```
trans.Value("Kwota") = 123.45
trans.Value("termin") = "2001-06-12"
trans.Value("stopa") = 0.12
```

3.12.3 Metody

3.12.3.1 FieldName

Składnia:

```
Function Zapis.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.12.3.2 FieldSize

Składnia:

```
Function Zapis.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.12.3.3 FieldType

Składnia:

```
Function Zapis.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.12.3.4 FieldDec

Składnia:

```
Function Zapis.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.12.3.5 FieldCount

Składnia:

```
Function Zapis.FieldCount( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.12.3.6 Insert

Składnia:

```
Function Transakcja.Insert() As Boolean
```

Opis:

Metoda dodaje nową transakcję do zapisu lub do dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli została utworzona nowa transakcja w zapisie lub dokumencie.

Przykład:

```
Dim t as transakcja
Set t = zap.Transakcja
If t.Insert then
    t.Value("termin") = "2001-06-28"
Endif
```

3.12.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.13 Rozliczenie

3.13.1 Opis

Obiekt **Rozliczenie** reprezentuje rozliczenie nowej, wcześniej utworzonej transakcji

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 całym dokumentem (sumą księgowañ na konto kontrahenta ze wszystkich zapisów), a nie z poszczególnymi zapisami na konta.

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 pojedynczymi zapisami na kontach rozrachunkowych.

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 jednego dokumentu dla faktur i RU oraz dla jednego zapisu dla pozostałych typów dokumentów może być utworzona najwyżej jedna nowa transakcja, odpowiadająca znacznikowi [X] *Nowa transakcja* na formacie wprowadzania rozrachunków w programie FK, oraz dowolna liczba rozliczeń istniejących transakcji.

W przypadku łączenia nowej transakcji z rozliczeniami transakcji dla jednego zapisu, kwota tego zapisu musi równać się sumie kwoty nowej transakcji oraz wszystkich rozliczeń.

Musi istnieć dokument i nowa transakcja, która ma być rozliczona.

3.13.2 Właściwości

3.13.2.1 Value

Składnia:

```
Rozliczenie.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w rozliczeniu.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Dostępne pola obiektu **Rozliczenie**:

- "kwota" Kwota rozliczająca nową transakcję
- "kwota_w" kwota rozliczenia w walucie – tylko dla kont i dokumentów walutowych
- "dokRozliczany" numer własny dokumentu, z którego nowa transakcja ma zostać rozliczona. Dla raportów kasowych jest to numer KP/KW pozycji raportu.

Dla rozliczeń walutowych wartość pola "kwota" musi być zgodna z wartością pola "kwota_w" przemnożoną przez kurs waluty w zapisie lub dokumencie, oraz waluta rozliczenia musi być taka sama jak waluta rozliczanej transakcji.

Rozliczenie może być dokonane wyłącznie dla istniejącej już nowej transakcji. Dokument tworzący rozliczaną nową transakcję może być w księgach albo w buforze.

Przykład

```
rozl.Value("Kwota") = 123.45  
rozl.Value("dokRozliczany") = "FA 06/01/00012"
```

3.13.3 Metody

3.13.3.1 FieldName

Składnia:

```
Function Rozliczenie.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.13.3.2 FieldSize

Składnia:

```
Function Rozliczenie.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.13.3.3 FieldType

Składnia:

```
Function Rozliczenie.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.13.3.4 FieldDec

Składnia:

```
Function Rozliczenie.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.13.3.5 FieldCount

Składnia:

```
Function Rozliczenie.FieldCount( Index As Variant) As integer
```


Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.13.3.6 MoveFirst

Składnia:

```
Function Rozliczenie.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego pierwszego rozliczenia w dokumencie.

Przykład:

```
If roz.MoveFirst then  
    MsgBox "Kwota=" + Cstr(roz.Value("kwota"))  
endif
```

3.13.3.7 MoveNext

Składnia:

```
Function Rozliczenie.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego kolejnego rozliczenia w dokumencie.

Przykład:

```
roz.MoveFirst  
While not roz.EOF then  
    MsgBox "Kwota =" + Cstr(roz.Value("kwota"))  
    roz.MoveNext  
Wend
```

3.13.3.8 Insert

Składnia:

```
Function Rozliczenie.Insert() As Boolean
```

Opis:

Metoda dodaje nowe rozliczenie do zapisu lub do dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli zostało utworzone kolejne rozliczenie w dokumencie.

Przykład:

```
If Not roz.Insert then  
    MsgBox "nowe rozliczenie nie może być utworzone"  
Endif
```

3.13.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.14 RejestrVAT

3.14.1 Opis

Obiekt **RejestrVAT** występuje w dwóch postaciach:

- Jako element obiektu **PDokument**, wówczas reprezentuje informację o rejestrach VAT związanych z danym dokumentem. Obiekt związany z dokumentem jest dostępny poprzez właściwość **Dokument.RejestrVAT**
- Jako samodzielny obiekt, wówczas umożliwia pobranie informacji o rejestrach VAT wielu dokumentów, wyselekcjonowanych według określonych kryteriów. Obiekt samodzielny jest tworzony przez deklarację New lub jawne wywołanie metody **CreateObject**.

W rejestrze zapisana jest informacja o kwotach brutto, netto i VAT oraz stawce VAT, względnie informacji o zwolnieniu od podatku, lub niepodlegające podatkowi VAT.

Obiekt **RejestrVAT** w danej chwili udostępnia informację z jednego, bieżącego rejestru. Dostęp do kolejnych rejestrów jest sekwencyjny, poprzez metody **MoveFirst** oraz **Movenext**.

3.14.2 Właściwości

3.14.2.1 EOF

Składnia:

```
Dim Rejestr As RejestrVAT
Rejestr.EOF As Boolean
```

Opis:

Właściwość **EOF** informuje, czy dany (kolejny) rejestr istnieje i został ustawiony, jako bieżący. Właściwość ta jest ustawiana przez następujące metody:

- MoveFirst
- MoveNext

Jest to właściwość tylko do odczytu.

Przykład:

```
Dim rej as RejestrVAT
Set rej = dok.RejestrVAT
rej.MoveFirst
While Not rej.EOF
    'robimy coś interesującego z tym obiektem
    Rej.MoveNext
Wend
```

3.14.2.2 Parent

Składnia:

```
Rejestr.Parent As Object
```

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **Dokument**, do którego należy dany rejestr.

W przypadku obiektu samodzielnego, niezwiązanego z dokumentem ta właściwość nie zwraca żadnego obiektu.

Przykład:

```
Dim r as rejestrVAT
Dim d as Dokument
...
set d = r.parent
```

```
d.Value("datadok") = "2001-07-10"
```

3.14.2.3 Value

Składnia:

```
Rejestr.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w rejestrze.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "rejVat" w bazie danych programu.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Przykład

```
rej.Value("netto") = 123.45  
rej("ABC") = 1
```

3.14.2.4 PozycjaVAT

Składnia:

```
Rejestr.PozycjaVAT As Object
```

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **PozycjaVAT** dla danego rejestru.

Przykład:

```
Dim r as rejestrVAT  
Dim p as PozycjaVAT  
...  
set p = r. PozycjaVAT  
p.Value("ue") = 1  
p.Value("usluga") = 1
```

3.14.2.5 Value

Składnia:

```
Rejestr.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w rejestrze.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "rejVat" w bazie danych programu.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Przykład

```
rej.Value("netto") = 123.45  
rej("ABC") = 1
```

3.14.3 Metody

3.14.3.1 Open

Składnia:

```
Function Rejestr.Open(Firma As BtDatabase, Ksiegi As Boolean, defId As long,  
okresOd As string, okresDo As string ) As Boolean
```

Opis:

Metoda jest dostępna tylko dla samodzielnego obiektu, niezwiązanego z dokumentem. Umożliwia wybranie i udostępnienie zbioru rejestrów VAT, według zadanych kryteriów.

parametry:

Firma – obiekt reprezentujący otwarte połączenie do bazy danych programu FK.

Ksiegi – Wartość logiczna określająca lokalizację dokumentów, z których chcemy pobrać rejestry. Wartość **True** oznacza księgi, wartość **False** oznacza bufor.

DefId – jest to identyfikator wybranej definicji rejestru. Wartość **zero** powoduje wybranie rejestrów ze wszystkich definicji.

OkresOd – Okres, od którego począwszy mają być wybrane dane z rejestrów VAT. Może być podany tylko rok i miesiąc, albo pełna data, z której znaczenie ma tylko rok i miesiąc (dzień jest ignorowany). Jeśli jako parametr zostanie podany pusty tekst lub data z rokiem równym **0000**, oznacza to wszystkie rejestry od początku pracy z programem.

OkresDo – Okres, do którego włącznie mają być wybrane dane z rejestrów VAT. Może być podany tylko rok i miesiąc, albo pełna data, z której znaczenie ma tylko rok i miesiąc (dzień jest ignorowany). Jeśli jako parametr zostanie podany pusty tekst lub data z rokiem równym **0000**, oznacza to wszystkie rejestry do ostatniego wprowadzonego do programu.

Przykład:

```
Dim dbase as New BtDataBase  
'inicjujemy połączenie z bazą danych  
'...  
'Deklarujemy RejestrVat jako obiekt samodzielny  
Dim rej as New RejestrVAT  
'wybieramy rejestry z dokumentów zaksięgowanych,  
'ze stycznia i lutego roku 2006  
Rej.Open( dbase, True, 12, „2006-01”, „2006-02”)  
rej.MoveFirst  
While Not rej.EOF  
    'robimy coś interesującego z tym obiektem  
    '...  
    'i pobieramy następny obbiekt  
    Rej.MoveNext  
Wend
```

3.14.3.2 FieldName

Składnia:

```
Function Rejestr.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.14.3.3 FieldSize

Składnia:

```
Function Rejestr.FieldSize ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.14.3.4 FieldType

Składnia:

```
Function rejestr.FieldType ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.14.3.5 FieldDec

Składnia:

```
Function Rejestr.FieldDec ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.14.3.6 FieldCount

Składnia:

```
Function Rejestr.FieldCount As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.14.3.7 MoveFirst

Składnia:

```
Function Rejestr.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego pierwszego rejestru w dokumencie, lub dla samodzielnego obiektu niezwiązanego z dokumentem powoduje ustawienie pierwszego rejestru z grupy wyselekcjonowanej poleceniem **Open**.

Przykład:

```
If rej.MoveFirst then  
    MsgBox "Kwota netto=" + Cstr(rej.Value("netto"))  
endif
```

3.14.3.8 MoveNext

Składnia:

```
Function Rejestr.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego kolejnego rejestru w dokumencie.

Przykład:

```
rej.MoveFirst  
While not rej.EOF then  
    MsgBox "Kwota netto=" + Cstr(rej.Value("netto"))  
    rej.MoveNext
```

3.14.3.9 Insert

Składnia:

```
Function Rejestr.Insert() As Boolean
```

Opis:

Metoda dodaje nowy rejestr do grupy zapisów lub do dokumentu. Metoda może być wywołana tylko dla obiektu związanego z **Dokumentem**.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli został utworzony kolejny rejestr w dokumencie.

Przykład:

```
If Not rej.Insert then
    MsgBox "nowy rejestr nie może być utworzony"
Endif
```

3.14.3.10 Delete

Składnia:

```
Function Rejestr.Delete() As Boolean
```

Opis:

Metoda usuwa bieżący rejestr z dokumentu. Metoda może być wywołana tylko dla obiektu związanego z **Dokumentem**.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli rejestr został usunięty.

Przykład:

```
If not rej.Delete() then
    MsgBox "rejestr nie może być usunięty"
Endif
```

3.14.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.15 PRejestr (Uwaga)

UWAGA! Klasa jest utrzymywana dla kompatybilności z wersjami wcześniejszymi niż 5.10. Ponieważ nie posiada pełnej funkcjonalności należy zamiast niej używać klasy RejestrVAT.

3.15.1 Opis

Obiekt **PRejestr** reprezentuje informację zapisaną w jednym rejestrze VAT (odpowiadającej jednej definicji rejestru w programie FK), oraz dla zakupów - dla jednego rodzaju zakupu A, B lub C.

W jednym rejestrze zapisana jest informacja o kwotach brutto, netto i VAT dla maksymalnie 4 stawek niezerowych oraz stawkach 0%, zwolnione od podatku, oraz niepodlegające podatkowi VAT.

Z jednym dokumentem mogą być związane maksymalnie 3 różne rejestry VAT, dla rejestrów zakupu dodatkowo podzielone na rodzaje zakupów.

Obiekt **PRejestr** umożliwia dostęp do jednego, bieżącego rejestru (lub rodzaju zakupu). Dostęp do kolejnych rejestrów jest sekwencyjny, poprzez metody **MoveFirst** oraz **Movenext**.

UWAGA. w obecnej implementacji obiekt integracji pozwala wyłącznie na tworzenie rejestrów dla dokumentów typu faktura i rachunek uproszczony. Nie jest możliwe tworzenie rejestrów VAT dla raportów kasowych. Jest to natomiast dostępne za pośrednictwem klasy RejestrVAT.

3.15.2 Właściwości

3.15.2.1 EOF

Składnia:

```
Rejestr.EOF As Boolean
```

Opis:

Właściwość **EOF** informuje, czy dany (kolejny) rejestr istnieje i został ustawiony, jako bieżący. Właściwość ta jest ustawiana przez następujące metody:

- MoveFirst
- MoveNext

Jest to właściwość tylko do odczytu.

Przykład:

```
Dim rej as PRejestr
Set rej = dok.rejestr
rej.MoveFirst
While Not rej.EOF
    'robimy coś interesującego z tym zapisem
    Zap.MoveNext
Wend
```

3.15.2.2 Parent

Składnia:

```
Rejestr.Parent As Object
```

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **Dokument**, do którego należy dany rejestr.

Przykład:

```
Dim r as PRejestr
Dim d as Dokument
...
set d = r.parent
d.Value("datadok") = "2001-07-10"
```

3.15.2.3 Value

Składnia:

```
Rejestr.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola w rejestrze.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "Rejestry" w bazie danych programu.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Przykład

```
rej.Value("netto1") = 123.45
rej("ABC") = 1
```

3.15.3 Metody

3.15.3.1 FieldName

Składnia:

```
Function Rejestr.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.15.3.2 FieldSize

Składnia:

```
Function Rejestr.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.15.3.3 FieldType

Składnia:

```
Function rejestr.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.15.3.4 FieldDec

Składnia:

```
Function Rejestr.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.15.3.5 FieldCount

Składnia:

```
Function Rejestr.FieldCount As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.15.3.6 MoveFirst

Składnia:

```
Function Rejestr.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącego pierwszego rejestru w dokumencie.

Przykład:

```
If rej.MoveFirst then  
    MsgBox "Kwota netto=" + Cstr(rej.Value("netto1"))  
endif
```

3.15.3.7 MoveNext

Składnia:

```
Function Rejestr.MoveNext() As Boolean
```


Opis:

Metoda powoduje ustawienie, jako bieżącego kolejnego rejestru w dokumencie.

Przykład:

```
rej.MoveFirst
While not rej.EOF then
    MsgBox "Kwota netto=" + Cstr(rej.Value("netto1"))
    rej.MoveNext
Wend
```

3.15.3.8 Insert

Składnia:

```
Function Rejestr.Insert() As Boolean
```

Opis:

Metoda dodaje nowy rejestr do grupy zapisów lub do dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli został utworzony kolejny rejestr w dokumencie.

Przykład:

```
If Not rej.Insert then
    MsgBox "nowy rejestr nie może być utworzony"
Endif
```

3.15.3.9 Delete

Składnia:

```
Function Rejestr.Delete() As Boolean
```

Opis:

Metoda usuwa bieżący rejestr z dokumentu.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli rejestr został usunięty.

Przykład:

```
If not rej.Delete() then
    MsgBox "rejestr nie może być usunięty"
Endif
```

3.15.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.16 DefDokum

3.16.1 Opis

Obiekt zapewnia dostęp do definicji dokumentów w programie FK. Poprzez ten obiekt realizowany jest dostęp zarówno do dokumentów predefiniowanych, jak również zdefiniowanych przez użytkownika

3.16.2 Właściwości

3.16.2.1 FieldCount

Składnia:

```
DefDokum.FieldCount As integer
```

Opis:

Właściwość zwraca liczbę wszystkich pól dostępnych poprzez obiekt **DefDokum**.

3.16.2.2 Value

Składnia:

```
DefDokum.Value ( Index as VARIANT ) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie dowolnego pola w definicji dokumentu.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "DefDokum" w bazie danych programu.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Jest to właściwość tylko do odczytu. Poprzez obiekt integracji nie mogą być zmienione dane definicji dokumentów

Przykład

```
dd.MoveTo ("DP")  
Msgbox dd.Value ("dNazwa")
```

3.16.3 Metody

3.16.3.1 Open

Składnia:

```
Function DefDokum.Open ( rok as Integer, firma as BtDatabase ) As Boolean
```

Opis:

Metoda inicjalizuje obiekt danymi wybranego roku obrachunkowego. Rok identyfikowany jest przez jego indeks. Pierwszy rok działalności ma indeks 0 (zero). Jako drugi parametr metody należy podać obiekt BtDatabase, reprezentujący otwarte połączenie do danych firmy

Przykład:

```
Dim db as new BtDatabase  
...  
Dim dd as new DefDokum  
dd.Open 0, db  
dd.MoveTo ("DP")  
MsgBox "Pełna nazwa dokumentu DP to " + dd.Value ("dNazwa")
```

3.16.3.2 MoveFirst

Składnia:

```
Function DefDokum.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie jako bieżącej pierwszej definicji dokumentu. Obiekt realizuje w danym momencie dostęp tylko do jednej, bieżącej definicji.

Przykład:

```
If dd.MoveFirst then  
    MsgBox "Nazwa=" + dd.Value ("dNazwa")  
endif
```

3.16.3.3 MoveNext

Składnia:

```
Function DefDokum.MoveNext () As Boolean
```

Opis:

Metoda powoduje ustawienie jako bieżącej kolejnej definicji dokumentu.

Wartość zwracana:

True – ustawiono kolejną definicję dokumentu

False – nie ma następnej definicji dokumentu

Przykład:

```
Dim ok as Boolean  
Ok = pp.MoveFirst  
While ok  
    MsgBox "nazwa=" + pp.Value("dNazwa")  
    Ok = pp.MoveNext  
Wend
```

3.16.3.4 MoveTo

Składnia:

```
Function DefDokum.MoveTo ( Skrot as String) As Boolean
```

Opis:

Metoda powoduje ustawienie jako bieżącej definicji dokumentu o podanym skrótce.

Przykład:

```
dd.MoveTo ("DP")  
MsgBox "Pełna nazwa dokumentu DP to " + dd.Value("dNazwa")
```

3.16.3.5 FieldName

Składnia:

```
Function DefDokum.FieldName ( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.16.3.6 FieldSize

Składnia:

```
Function DefDokum.FieldSize ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.16.3.7 FieldType

Składnia:

```
Function DefDokum.FieldType ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.16.3.8 FieldDec

Składnia:

```
Function DefDokum.FieldDec ( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.16.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.17 DefRejVAT

3.17.1 Opis

Obiekt zapewnia dostęp do definicji rejestrów VAT zdefiniowanych w programie FK.

3.17.2 Właściwości

3.17.2.1 FieldCount

Składnia:

```
DefRejVAT.FieldCount As integer
```

Opis:

Właściwość zwraca liczbę wszystkich pól dostępnych poprzez obiekt **DefRejVAT**.

3.17.2.2 Value

Składnia:

```
DefRejVAT.Value ( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie dowolnego pola w definicji rejestru.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "defRej" w bazie danych programu. Dodatkowo zdefiniowane jest pole „Naliczany”.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Jest to właściwość tylko do odczytu. Poprzez obiekt integracji nie mogą być zmienione dane definicji rejestru.

Przykład

```
dr.MoveFirst()  
Msgbox dr.Value("rNazwa")
```

3.17.3 Metody

3.17.3.1 Open

Składnia:

```
Function DefRejVAT.Open(firma as BtDatabase ) As Boolean
```

Opis:

Metoda inicjalizuje obiekt danymi dotyczącymi definicji wszystkich rejestrów VAT dla firmy. Jako parametr metody należy podać obiekt BtDatabase, reprezentujący otwarte połączenie do danych firmy

Przykład:

```
Dim db as new BtDatabase
...
Dim dr as new DefRejVAT
dr.Open db
dr.MoveFirst()
MsgBox "Nazwa pierwszej definicji to " + dr.Value("rNazwa")
```

3.17.3.2 MoveFirst

Składnia:

```
Function DefRejVAT.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie jako bieżącej pierwszej definicji rejestru. Obiekt realizuje w danym momencie dostęp tylko do jednej, bieżącej definicji.

Przykład:

```
If dr.MoveFirst then
    MsgBox "Nazwa=" + dr.Value("rNazwa")
endif
```

3.17.3.3 MoveNext

Składnia:

```
Function DefRejVAT.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej kolejnej definicji rejestru.

Wartość zwracana:

True – ustawiono kolejną definicję rejestru

False – nie ma następnej definicji rejestru

Przykład:

```
Dim ok as Boolean
Ok = dr.MoveFirst
While ok
    MsgBox "nazwa=" + dr.Value("rNazwa")
    Ok = dr.MoveNext
Wend
```

3.17.3.4 MoveTo

Składnia:

```
Function DefRejVAT.MoveTo( Id as Long) As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej definicji rejestru o podanym identyfikatorze.

Przykład:

```
dr.MoveTo(1)
MsgBox "nazwa definicji o id=1 to " + dr.Value("rNazwa")
```

3.17.3.5 FieldName

Składnia:

```
Function DefRejVAT.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.17.3.6 FieldSize

Składnia:

```
Function DefRejVAT.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.17.3.7 FieldType

Składnia:

```
Function DefRejVAT.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.17.3.8 FieldDec

Składnia:

```
Function DefRejVAT.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.17.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.18 DefRej (Uwaga)

3.18.1 Opis

Obiekt tej klasy zapewnia dostęp do definicji rejestrów zdefiniowanych w programie FK dla konkretnego roku.

UWAGA! Klasa ta została zaprojektowana do współpracy z wersjami FK wcześniejszymi niż 5.10. Jest utrzymywana dla kompatybilności wstecz, ale nie zapewnia pełnej funkcjonalności bieżącej postaci rejestrów VAT. w kolejnych wersjach modułu Integracji nie będzie utrzymywana. w jej miejsce należy używać obiektów klasy DefRejVAT.

3.18.2 Właściwości

3.18.2.1 FieldCount

Składnia:

```
DefRej.FieldCount As integer
```

Opis:

Właściwość zwraca liczbę wszystkich pól dostępnych poprzez obiekt **DefRej**.

3.18.2.2 Value

Składnia:

```
DefRej.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie dowolnego pola w definicji rejestru.

Wszystkie pola są identyfikowane poprzez Index, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiadają bezpośrednio zawartości tabeli "DefRej" w bazie danych programu.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Jest to właściwość tylko do odczytu. Poprzez obiekt integracji nie mogą być zmienione dane definicji rejestru.

Przykład

```
dr.MoveFirst()  
Msgbox dr.Value("rNazwa")
```

3.18.3 Metody

3.18.3.1 Open

Składnia:

```
Function Defrej.Open( rok as Integer, firma as BtDatabase ) As Boolean
```

Opis:

Metoda inicjalizuje obiekt danymi wybranego roku obrachunkowego. Rok identyfikowany jest przez jego indeks. Pierwszy rok działalności ma indeks 0 (zero). Jako drugi parametr metody należy podać obiekt BtDatabase, reprezentujący otwarte połączenie do danych firmy

Przykład:

```
Dim db as new BtDatabase  
...  
Dim dr as new DefRej  
dr.Open 0, db  
dr.MoveFirst()  
MsgBox "Nazwa pierwszej definicji to " + dr.Value("rNazwa")
```

3.18.3.2 MoveFirst

Składnia:

```
Function DefRej.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej pierwszej definicji rejestru. Obiekt realizuje w danym momencie dostęp tylko do jednej, bieżącej definicji.

Przykład:

```
If dr.MoveFirst then  
    MsgBox "Nazwa=" + dr.Value("rNazwa")  
endif
```

3.18.3.3 MoveNext

Składnia:

```
Function DefRej.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej kolejnej definicji rejestru.

Wartość zwracana:

True – ustawiono kolejną definicję rejestru

False – nie ma następnej definicji rejestru

Przykład:

```
Dim ok as Boolean
Ok = dr.MoveFirst
While ok
    MsgBox "nazwa=" + dr.Value("rNazwa")
    Ok = dr.MoveNext
Wend
```

3.18.3.4 MoveTo

Składnia:

```
Function Defrej.MoveTo( Id as Long) As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej definicji rejestru o podanym identyfikatorze.

Przykład:

```
dr.MoveTo(1)
MsgBox "nazwa definicji o id=1 to " + dr.Value("rNazwa")
```

3.18.3.5 FieldName

Składnia:

```
Function Defrej.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.18.3.6 FieldSize

Składnia:

```
Function Defrej.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.18.3.7 FieldType

Składnia:

```
Function Defrej.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.18.3.8 FieldDec

Składnia:

```
Function Defrej.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.18.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.19 PlanKont

3.19.1 Opis

Obiekt umożliwia dostęp do definicji kont wybranego roku obrachunkowego. Poprzez ten obiekt można wyłącznie odczytywać informacje, nie można zmieniać definicji ani dodawać lub usuwać kont.

3.19.2 Właściwości

3.19.2.1 FieldCount

Składnia:

```
PlanKont.FieldCount As integer
```

Opis:

Właściwość zwraca liczbę wszystkich pól dostępnych poprzez obiekt **PlanKont**.

3.19.2.2 Value

Składnia:

```
PlanKont.Value ( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie dowolnego pola w definicji konta.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól. Nazwy pól oraz ich typy i rozmiary odpowiada z grubsza atrybutom zdefiniowanej tabeli w bazie. W celu otrzymania pełnej listy dla aktualnie używanej wersji można użyć iteracji z przykładu z początku dokumentu. Dodatkowo dołożone jest spreparowane pole "SubjectType", w którym jest przechowany numer kartoteki zgodny z konwencją numeracji numeratora SubjectType. Jest on wypełniany, gdy do konta podczepiona jest kartoteka.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Jest to właściwość tylko do odczytu. Poprzez obiekt integracji nie mogą być zmienione dane definicji kont

Przykład

```
pk.MoveTo("100-1")  
Msgbox pk.Value("Nazwa")
```

3.19.2.3 Listek

Składnia:

```
PlanKont.Listek As Boolean
```

Opis:

Właściwość zwraca informację, czy podane konto jest ostatnim poziomem analityki. Oznacza to równocześnie, że na dane konto może być zaksięgowany zapis.

3.19.2.4 Rozrachunkowe

Składnia:

```
PlanKont.Rozrachunkowe As Boolean
```

Opis:

Właściwość zwraca informację, czy na podanym koncie mogą być prowadzone rozrachunki. Jest to spełnione wówczas, gdy konto jest jawnie oznaczone w planie kont, jako rozrachunkowe albo, gdy na ostatnim poziomie analityki jest podpięta kartoteka kontrahentów, pracowników albo urzędów.

3.19.3 Metody

3.19.3.1 Open

Składnia:

```
Function PlanKont.Open( rok as Integer, firma as BtDatabase ) As Boolean
```

Opis:

Metoda inicjalizuje obiekt danymi wybranego roku obrachunkowego. Rok identyfikowany jest przez jego indeks. Pierwszy rok działalności ma indeks 0 (zero). Jako drugi parametr metody należy podać obiekt BtDatabase, reprezentujący otwarte połączenie do danych firmy

Przykład:

```
Dim db as new BtDatabase
...
Dim pk as new PlanKont
pk.Open 0, db
pk.MoveTo("201-1")
MsgBox "Pełna nazwa konta 201-1 to " + pk.Value("Nazwa")
```

3.19.3.2 MoveFirst

Składnia:

```
Function PlanKont.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej definicji pierwszego konta w planie kont, czyli syntetyki o najniższym numerze. Obiekt realizuje w danym momencie dostęp tylko do jednej, bieżącej definicji.

Przykład:

```
If pk.MoveFirst then
    MsgBox "Nazwa=" + pk.Value("Nazwa")
endif
```

3.19.3.3 MoveNext

Składnia:

```
Function PlanKont.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej kolejnej definicji konta.

Wartość zwracana:

True – ustawiono kolejną definicję konta

False – nie ma następnej definicji konta

Kolejność definicji kont w obiekcie **PlanKont** jest zgodna z kolejnością ułożenia kont w planie kont programu FK.

Przykład:

```
Ok = pk.MoveFirst
While ok
    MsgBox "nazwa=" + pk.Value("Nazwa")
    Ok = pk.MoveNext
Wend
```

3.19.3.4 MoveTo

Składnia:

```
Function PlanKont.MoveTo( Konto as String) As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej definicji konta o podanym numerze.

Przykład:

```
pk.MoveTo("201-1")  
MsgBox "Nazwa konta 201-1 to " + pk.Value("Nazwa")
```

3.19.3.5 FieldName

Składnia:

```
Function DefDokum.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.19.3.6 FieldSize

Składnia:

```
Function DefDokum.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.19.3.7 FieldType

Składnia:

```
Function DefDokum.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.19.3.8 FieldDec

Składnia:

```
Function DefDokum.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.19.4 Zdarzenia

Obiekt nie generuje zdarzeń

3.20 Rozrachunki

3.20.1 Opis

Obiekt zapewnia dostęp do danych o rozrachunkach.

3.20.2 Właściwości

3.20.2.1 BtDatabase

Składnia:

```
Rozrachunki.BtDatabase As BtDatabase
```

Opis:

Właściwość umożliwia pobranie aktywnego połączenia, na którym pracuje obecnie obiekt.

Przykład

```
set database = roz.BtDatabase
```

3.20.2.2 Value

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

Dostępne pola:

- Dane transakcji
- idTrans
- idRozl
- Dane dokumentu
- termin
- dokData
- dokId
- dokRok
- bufor
- dokSkrot
- dokNumer
- dokSymbol
- dokDataOkr
- Dane kwotowe
- kwota
- kwotaWal
- waluta
- kurs
- kursEuro
- kwotaTrans
- kwotaTransWal
- Stan transakcji
- rozliczona
- stan
- Dane kontrahenta
- subTyp

- pozycja
- subName
- Dabe konta
- synt
- poz1, poz2, poz3, poz4, poz5
- Pozostałe dane
- strona
- Uwagi
- count

Dodatkowe pola dla linii Symfonia ERP (od wersji 2010):

- dataWpr , data wprowadzenia
- dataMod , data modyfikacji
- Uld , guid dokumentu lub zapisu – tej encji, z którą związane rozrachunek
- Removed , wartość niezerowa oznacza usunięty rozrachunek
- Related , enum z jaką daną związane rozrachunek, 1-dokument, 2 – zapis, 3 – BO/OR.

3.20.2.3 LastId

Opis:

Tylko w linii Symfonia ERP. Właściwość umożliwia ustawienie wartości progowej id, od której będzie można wyszukiwać rozrachunki z pomocą metody LoadNew().

3.20.2.4 LastModification

Opis:

Tylko w linii Symfonia ERP. Właściwość umożliwia ustawienie wartości progowej daty (wprowadzenia, modyfikacji), od której będzie można wyszukiwać rozrachunki z pomocą metody LoadNew().

3.20.2.5 SimpleInfo

Opis:

Tylko w linii Symfonia ERP. Właściwość do ustawienia umożliwia zlecenie pobierania tylko ograniczonej (Simple) informacji z rozrachunków (wtedy przekazać należy wartość różną od zera). Wpływa na funkcjonowanie metody Load() i LoadNew() w taki sposób, że nie są pobierane żadne dodatkowe informacje z kartoteki podmiotów dla rozrachunku. Pozwala to w wydajniejszym trybie odzyskiwać dane z bazy.

3.20.3 Parametry wyszukiwania rozrachunków

SubjectType – typ podmiotu kontrahent, pracownik, urząd, ...

pozycja – identyfikator w FK kontrahenta, pracownika, urzędu, ...

DataOd – transakcje od daty

DataDo – transakcje do daty

BezDaty – czy pokazać transakcje bez określonego terminu

NaDzien – stan rozrachunków na dzień

DocumentName – nazwa dokumentu, względnie guid operacji bankowej (od wersji 2013 Symfonii)

Type – typ, należności, zobowiązania, ...

StanType – stan rozrachunku, rozliczony, nierozliczony, przeterminowany, ...

KwotaOd – transakcje o kwocie większej niż

KwotaDo – transakcje o kwocie mniejszej niż

Waluta – transakcje w określonej walucie

SetKonto - metoda ustawia konto, według którego mają być wyszukiwane transakcje. Nieużywane analityki powinny być ustalone na 0.

3.20.4 Metody

3.20.4.1 FieldCount, FieldName, FieldSize, FieldType

Opis:

Metody te są opisane w części opisującej elementy wspólne.

3.20.4.2 MoveFirst, MoveNext, MovePrev, MoveLast

Składnia:

```
Function Rozrachunki.MoveFirst() As Int
```

Opis:

Metody pozwalają na iterowanie się po liście wyszukanych rozrachunków lub agregatów rozrachunków.

Przykład:

```
Dim nRet as Int
nRet = roz.MoveFirst
while nRet = 0
    ...
    nRet = roz.MoveNext
wend
```

3.20.4.3 Open

Składnia:

```
Function Rozrachunki.Open(firma as BtDatabase )
```

Opis:

Metoda inicjalizuje obiekt danymi firmy. Jako parametr metody należy podać obiekt BtDatabase, reprezentujący otwarte połączenie do danych firmy.

Przykład:

```
Dim db as new BtDatabase
...
Dim roz As New Rozrachunki
roz.Open db
```

3.20.4.4 ClearParams

Składnia:

```
Sub Rozrachunki.ClearParams()
```

Opis:

Metoda ClearParams czyści parametry wyszukiwania. Nie przestawia wartości atrybutów obsługiwanych przez ClearNewParams.

3.20.4.5 ClearNewParams

Składnia:

```
Sub Rozrachunki.ClearNewParams()
```

Opis:

Tylko dla wersji Symfonia ERP. Metoda ClearNewParams czyści wyłącznie atrybuty wyszukiwania LastId, LastModification. z jej pomocą można odpytywać o ostatnio zmienione rozrachunki używając LoadNewParams.

3.20.4.6 ValidateParams

Składnia:

```
Function Rozrachunki.ValidateParams() as Boolean
```

Opis:

Metoda ValidateParams sprawdza poprawność parametrów. Gdy nie są prawidłowe zwraca FALSE.

3.20.4.7 OrderBy, GroupBy

Składnia:

```
Sub GroupBy(group As String)  
Sub OrderBy(sort As String)
```

Opis:

Metody ustawiają kryteria grupowania/sortowania zwracanych danych. Parametr powinien zawierać pole dostępne w FieldName (dostępne pola podane są przy właściwości Value).

Przykład:

```
roz.GroupBy("pozycja") ' grupowanie po podmiocie  
' aby sortować po większej liczbie kolumn należy użyć powtórzenia instrukcji  
' w pożądanej kolejności  
roz.OrderBy("pozycja") ' sortowanie po numerze kartotekowym podmiotu  
roz.OrderBy("kwota") ' sortowanie po kwocie
```

3.20.4.8 Load

Składnia:

```
Sub Load()
```

Opis:

Metoda wyszukuje (napędza listę) dane rozrachunków według zadanych kryteriów, dodatkowo grupuje i sortuje dane.

3.20.4.9 LoadNew

Składnia:

```
Sub LoadNew()
```

Opis:

Tylko dla wersji Symfonia ERP. Podobnie jak metoda Load(), wyszukuje dane rozrachunków według zadanych kryteriów, dodatkowo grupuje i sortuje dane. Są to jednak dane o nowych (świeżych) rozrachunkach, bowiem specyfiką tej metody jest respektowanie dodatkowych warunków:

- rekordów o id większym od przekazanego atrybutu LastId,

- o dacie modyfikacji lub wprowadzenia późniejszej niż ustawiony LastModification

Ponadto w przygotowanej liście danych znajdują się również informacje o usuniętych rozrachunkach spełniających warunki filtrowania.

Metoda ta jest wprowadzona od wersji 2010 i jest wykorzystywana w nowej synchronizacji rozrachunków z modulem Handel.

3.21 Rozliczenia

3.21.1 Opis

Obiekt zapewnia dostęp do danych o rozliczeniach konkretnej transakcji.

3.21.2 Właściwości

3.21.2.1 Value

Opis:

Właściwość jest opisana w części dotyczącej elementów wspólnych

Dostępne pola:

- Dane transakcji
- idTrans
- idRozl
- Dane dokumentu
- data
- dokId
- dokRok
- bufor
- dokSkrot
- dokNumer
- dokSymbol
- dokDataOkr
- dokPozNum
- Dane kwotowe
- kwota
- kwotaWal
- waluta
- kurs
- kursEuro
- Pozostałe dane
- strona

- Uwagi

3.21.2.2 NaDzien

Składnia:

```
Rozliczenia.NaDzien As Date
```

Opis:

Właściwość umożliwia ustawienie stanu rozrachunków na dzień dla danej transakcji.

3.21.3 Metody

3.21.3.1 FieldCount

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.21.3.2 FieldName

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.21.3.3 FieldSize

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.21.3.4 FieldType

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.21.3.5 MoveFirst, MoveNext

Składnia:

```
Function Rozliczenia.MoveFirst() As Int
```

Opis:

Metody pozwalają na iterowanie się po liście wyszukanych rozliczeń.

Przykład:

```
Dim nRet as Int
nRet = rozlicz.MoveFirst
while nRet = 0
    ...
    nRet = rozlicz.MoveNext
Wend
```

3.21.3.6 Open

Składnia:

```
Sub Rozliczenia.Open(pBtDatabase As BtDatabase, nIdTrans As Long, nIdRozl As Long)
```

Opis:

Metoda inicjalizuje obiekt danymi firmy oraz wyszukuje rozliczenia dla przekazanych parametrów (identyfikatora transakcji i identyfikatora rozliczenia). Parę parametrów można pobrać z danych zwróconych przez obiekt Rozrachunki.

Przykład:

```
Dim db as new BtDatabase
...
Dim roz As New Rozrachunki
Dim rozlicz As New Rozliczenia
idTrans = roz.Value("idTrans")
idRoz = roz.Value("idRozl")
...
rozlicz.NaDzien = roz.NaDzien
rozlicz.Open db, idTrans, idRozl
```

3.22 DocErrors

3.22.1 Opis

Obiekt udostępnia informacje o błędach, uchybieniach, niespójności dokumentu, po wywołaniu walidacji lub zapisie dokumentu.

3.22.2 Właściwości

3.22.2.1 Errors

Składnia:

```
DocErrors.Errors As Long
```

Opis:

Informuje o występującym błędzie/błędach dokumentu (tylko tych o najwyższym poziomie), 0 gdy nie ma błędów.

3.22.2.2 Description

Składnia:

```
DocErrors.Description (Index As Long) as String
```

Opis:

Udostępnia tekstową informację o przyczynie błędu o podanym indeksie (numerowany od 0)

3.22.3 Metody

3.22.3.1 Count

Składnia:

```
Function Count() As Long
```

Opis:

Metoda zwraca liczbę błędów wszystkich poziomów z dokumentu po wykonaniu walidacji.

3.22.3.2 Type

Składnia:

```
Function Type( Index As Long ) As Long
```

Opis:

Metoda zwraca poziom błędu o zadanym indeksie:

0 – komentarz. Niespójność, którą obiekt Integracji może sam naprawić/usunąć.

1 – ostrzeżenie. o nieprawidłowościach, które jednak można usunąć później, po przyjęciu dokumentu.

2 – błąd. Dowolny, który uniemożliwi zaksięgowanie, niekiedy także zapis dokumentu do bazy.

Przykład (VBScript) :

```
Dim counter, mymessage
dim Errlist
set Errlist = CreateObject("Mxdokfk.DocErrors")

counter = 0
while counter < Errlist.count
    mymessage = "poziom błędu = " & Errlist.Type(counter)
    mymessage = mymessage + ", opis : " & Errlist.Description(counter)
    output mymessage ' funkcja zapisująca gdziekolwiek
    counter = counter+1
wend
```

3.23 InfoKonta

3.23.1 Opis

Obiekt zapewnia dostęp do odczytu obrotów kont poszczególnych lat, również z buforem.

3.23.2 Właściwości

3.23.2.1 Typ

Oznacza typ konta ze względu na jego właściwości dla bilansu i przyjmuje jedną z wartości:

- Dla kont pozabilansowych = 0,
- Dla kont bilansowych = 1,
- Dla kont wynikowych = 2.

Są to wartości widoczne w oknie planu kont.

3.23.2.2 Podtyp

Specyfikuje kategorię w ramach typu konta i przyjmuje jedną z wartości:

Zwykłe =0,

Rozrachunkowe = 1, (tylko dla kont bilansowych)

Szczególne = 2

3.23.2.3 Nazwa

Jest to Nazwa konta wpisana podczas definiowania konta lub nazwa elementu kartoteki, jeśli wartość analityki na ostatnim poziomie jest elementem z kartoteki.

3.23.2.4 Numer

Właściwość typu string przechowuje numer konta (w postaci tekstowej myślniki oddzielają kolejny poziom analityczny)

3.23.2.5 Flagi

Właściwość udostępnia dodatkowe znaczniki dla konta. Wartość jest sumą logiczną bitowych wartości:

- Konto z kontrolą zerowego Salda = 0x02

- Konto wieloletnie = 0x04
- Konto walutowe = 0x80

3.23.2.6 Listek

Właściwość udostępnia informację czy konto znajduje się na ostatnim poziomie analityki; 1 – tak.

3.23.2.7 ZBuforem

Znacznik uwzględniania zapisów będących w buforze. Wartości zaksięgowane są zawsze pobierane w naliczaniu. Właściwość można ustawiać. Wartość 1 oznacza uwzględnić.

3.23.2.8 Ma

Właściwość o typie Infopoz, służąca do odbierania informacji o kwotach obrotów za poszczególne okresy. Obiekt dla strony Ma zawiera kwoty konta dla bilansu otwarcia, obrotów konta narastająco w poszczególnych okresach roku obrachunkowego, oraz aktualny stan konta.

3.23.2.9 Wn

Właściwość o typie Infopoz, służąca do odbierania informacji o kwotach obrotów za poszczególne okresy. Obiekt dla strony Winien zawiera kwoty konta dla bilansu otwarcia, obrotów konta narastająco w poszczególnych okresach roku obrachunkowego, oraz aktualny stan konta.

3.23.3 Metody

3.23.3.1 Pierwsze

Składnia:

```
Pierwsze (TylkoIstnieje as long, SyntAnal as long, TypKonta as long) as String
```

Polecenie zwraca w postaci tekstowej numer pierwszego konta planu kont spełniającego zadane parametry. Zwraca pusty ciąg, jeśli nie można znaleźć żadnego konta. w przypadku wystąpienia błędu w trakcie uzyskiwania informacji można uzyskać jego opis.

TylkoIstnieje - parametr określający rodzaj iteracji:

- 0 - zwraca kolejno wszystkie pozycje z kartoteki kontrahentów;
- 1 - zwraca tylko te pozycje kartoteki, dla których było jakiekolwiek księgowanie
- 2 - zwraca tylko te pozycje kartoteki, dla których były jakiekolwiek zapisy

SyntAnal: określa, jakie konta zostaną zwrócone;

- 1 - tylko konta syntetyczne
- 0 - konta analityczne w tym syntetyki bez podpiętych kont analitycznych
- 1 - tylko konta analityczne z pominięciem syntetyk bez podpiętych kont analitycznych

TypKont: określa typ kont zwracanych;

- 1 - wszystkie konta
- 0 - konta pozabilansowe
- 1 - konta bilansowe
- 2 - konta wynikowe

3 - konta bilansowe i wynikowe

3.23.3.2 Kolejne

Polecenie zwraca w postaci tekstowej numer kolejnego konta planu kont spełniającego zadane parametry wyspecyfikowane w ostatnim wywołaniu Pierwsze() lub PierwszeOd(). Zwraca pusty ciąg jeśli nie można znaleźć już żadnego konta. w przypadku wystąpienia błędu w trakcie uzyskiwania informacji można uzyskać jego opis.

Składnia:

```
Kolejne () as String
```

3.23.3.3 PierwszeOd

Składnia:

```
PierwszeOd (kontoPocztkowe As String, TylkoIstnieje as long, SyntAnal as long, TypKonta as long) as String
```

Polecenie zwraca w postaci tekstowej numer pierwszego konta planu kont spełniającego zadane parametry. Zwraca pusty ciąg, jeśli nie można znaleźć żadnego konta. w przypadku wystąpienia błędu w trakcie uzyskiwania informacji można uzyskać jego opis. Polecenie różni się tym od metody Pierwsze(), że można w pierwszym parametrze podać, od którego konta zaczyna się iterowanie.

3.23.3.4 IdPK

Składnia:

```
IdPk () as long
```

Polecenie zwraca Identyfikator rekordu definiującego konto w planie kont.

3.23.3.5 Info

Składnia:

```
Info(nrKonta As String)
```

Wypełnia informacje o koncie podanym, jako argument w postaci tekstowej.

Uwaga - nie jest wypełniane InfoPoz

3.23.3.6 Konto

Składnia:

```
Konto () as long()
```

Polecenie zwraca numer konta w tablicy: numer syntetyki i numery kolejnych analityk. w numerze konta pierwszy element zawiera numer syntetyki, następne elementy zawierają numery kolejnych analityk.

3.23.3.7 Obroty

Składnia:

```
Obroty (nrkonta as String)
```

Polecenie wypełnia obiekty InfoPoz informacją o kwotach bilansu otwarcia, obrotach oraz aktualny stan konta. Wartości te można pobrać następnie przez właściwości Ma, Wn.

3.23.3.8 Open

Składnia:

```
Open(Firma As BtDatabase) As Long
```

Polecenie tworzy obiekt korzystając z istniejącego połączenia w celu dalszej obróbki.

3.23.3.9 Persaldo

Składnia:

PerSaldo (nrkonta as String)

Polecenie wypełnia obiekt Infopoz informacją o kwotach na koncie liczonych, jako suma sald (saldo dwustronne, persaldo) kont na ostatnim poziomie analityki dla konta podanego, jako argument w postaci tekstowej.

3.23.3.10 Skrot

Składnia:

Skrot() As String

Polecenie zwraca skróconą nazwę konta.

3.23.3.11 Słownik

Składnia:

Słownik() As Long()

Polecenie zwraca tablicę informującą o słownikach analitycznych na poszczególnych poziomach analityki. Wartość niezerowa elementu tablicy oznacza odpowiedni słownik lub kartotekę na danym poziomie analityki. Wartości dla Symfonii są następujące:

- 1 - kartoteka pracowników
- 2 - kartoteka kontrahentów
- 3 - kartoteka rachunków bankowych
- 4 - kartoteka urzędów
- 100-999 - kolejne słowniki zdefiniowane przez użytkownika

Wartości dla wersji Symfonia ERP to [id] z tabeli słowników podsystemu SCommon.

Przykład (VBscript) :

```
' db to obiekt typu BtDatabase
'wsc jest obiektem zawiadującym tworzeniem obiektów, pisaniem do logów

db.firmaInfo.SetRok 2      'indeks roku =2 otwartej bazy

wsc.setobj "Infokonta", sbj  'zadeklarowanie obiektu
sbj.Open db                  ' otwarcie w kontekście 3-ego roku bazy
ps = sbj.PierwszeOd( "100", 1, 0, 3 )
sbj.Zbuforem = 1
while not ps = ""
    s = sbj.idpk & ",typ:" & sbj.typ & ":numer:" & sbj.numer
    wsc.log s
    sbj.Obroty()
    if sbj.wn.mc( 0 ) > 0 then
        s = "winien: " & sbj.wn.mc( 0 ) + sbj.wn.bo & "; " & sbj.wn.cur
        wsc.log s
    end if
    ps = sbj.Kolejne
wend
```

3.24 InfoPoz

3.24.1 Opis

Obiekt współpracujący z obiektem InfoKonta, przechowuje informacje o kwotach wchodzących na określoną stronę konta. Zawiera kwoty zarówno z bilansu otwarcia, jak też narastająco dla poszczególnych miesięcy roku obrachunkowego oraz aktualny stan konta (ustawionego w obiekcie InfoKonta).

3.24.2 Właściwości

3.24.2.1 BO

Właściwość oddaje kwotę bilansu otwarcia, typu double.

3.24.2.2 Mc(int okres)

Oddaje kwotę dla podanego okresu sprawozdawczego narastająco, typu double.

3.24.2.3 Cur

Oddaje kwotę aktualną (stan na dzień dzisiejszy), typu double.

3.25 PozycjaVAT

3.25.1 Opis

Obiekt **PozycjaVAT** pozwala na poprawną ewidencję usług dla rejestrów VAT. Za pomocą obiektu można poprawnie ewidencjonować usługi w rejestrach i po zapisaniu odczytywać zapisane usługi. Do poprawnego zaewidencjonowania danych o usługach dla rejestrów integracja musi otrzymać następujące dane:

Stawka	– stawka, jaka została opodatkowana usługa
Stawka_pl	– stawka, jaka zostałaby opodatkowana usługa przy sprzedaży na terenie kraju
Nazwa	– Nazwa usługi
Wartosc	– wartość usługi (jaka część wartości z rejestru dotyczy usługi)
Usługa	– czy pozycja usług/czy towarów
UE	– czy sprzedaż na terenie Unii Europejskiej czy poza

Algorytm przyjmowania usług działa następująco. Sprawdza czy rejestr (parent) ma ustawione znaczniki usług i ue zgodnie z pozycjavat. Jeżeli nie dzieli rejestr w celu dodania wpisu o zaznaczonych znacznikach tak jak pozycjavat. Pozycje niebędące usługami będą pomijane i nie zostaną zaewidencjonowane, ustawodawca nie wymaga takiej ewidencji. Jeżeli powyższe pola zostaną wypełnione poprawnie integracja poprawnie zaewidencjonuje usługi. w przypadku, gdy zapis w rejestrze nie będzie się zgadzał z pozycjamivat zostaną zgłoszone błędy informujące, co jest błędnego. Analiza pozycjavat realizowana jest podczas zapisu dokumentu.

Obiekt **PozycjaVAT** został rozbudowany o obsługę kursów dla podatków VAT, CIT/PIT występujących dla dokumentów faktur walutowych VAT. Dla pozycji VAT z kursami jest inna interpretacja własności, opis własności używanych przy wprowadzaniu kursów i ich znaczenie jest podany w następnym rozdziale.

3.25.2 Własności

3.25.2.1 Value

Składnia:

```
pozycjavat.Value( Index as VARIANT) As VARIANT
```

Opis:

Właściwość **Value** umożliwia pobranie lub ustawienie wartości dowolnego pola pozycjavat.

Wszystkie pola są identyfikowane poprzez **Index**, który może być nazwą tekstową pola (wielkość liter nie ma znaczenia), lub indeksem (numerem kolejnym liczonym od 0) na liście pól.

Jest to domyślna właściwość obiektu, więc w językach skryptowych (VB) może być pominięta.

Pola dostępne/znaczenie:

Id	– dla usług i kursów id usługi zapisanej w bazie
RokId	– dla usług i kursów indeks roku
DokId	– dla usług i kursów id dokumentu z rejestrem
ExtId	– dla usług id rejestru, do którego przypisana jest usługa – dla kursów numer tabeli kursów
TypExt	– dla usług typ wiązania zewnętrznego do rejestru 78 – dla kursów podatków VAT = 1, CIT/PIT = 2
Stawka	– stawka (po zapisie zerowane)
Stawka_pl	– stawka przy sprzedaży na terenie kraju (po zapisie zerowane)
Nazwa	– opis (dla usług nazwa usługi) – dla kursów (VAT, CIT/PIT) skrót waluty
Wartosc	– wartość (dla usług wartość usługi) – dla kursów (VAT, CIT/PIT) wartość kursu
Usługa	– czy pozycja usług
UE	– czy sprzedaż na terenie Unii Europejskiej
Data	– data tabeli kursów dla podatków VAT, CIT/PIT

Przykład

```
pozycjavat.Value("ue") = 1
```

3.25.2.2 Parent

Składnia:

```
Pozycjavat.Parent As Object
```

Opis:

Właściwość tylko do odczytu. Zwraca obiekt typu **RejestrVAT**, do którego należą dane pozycjevat.

Przykład:

```
Dim p as PozycjaVAT
Dim r as RejestrVAT
...
set r = p.parent
r.Value("stawka") = "22"
```


3.25.3 Metody

3.25.3.1 FieldName

Składnia:

```
Function Pozycjivat.FieldName( Index As integer ) As String
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.25.3.2 FieldSize

Składnia:

```
Function Pozycjivat.FieldSize( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.25.3.3 FieldType

Składnia:

```
Function Pozycjivat.FieldType( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.25.3.4 FieldDec

Składnia:

```
Function Pozycjivat.FieldDec( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.25.3.5 FieldCount

Składnia:

```
Function Pozycjivat.FieldCount( Index As Variant) As integer
```

Opis:

Metoda jest opisana w części opisującej elementy wspólne

3.25.3.6 MoveFirst

Składnia:

```
Function Pozycjivat.MoveFirst() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej pierwszej pozycjivat dla rejestruvat.

Przykład:

```
If Pozycjivat.MoveFirst then  
    MsgBox "opis=" + Cstr(Pozycjivat.Value("opis"))  
endif
```

3.25.3.7 MoveNext

Składnia:

```
Function Pozycjivat.MoveNext() As Boolean
```

Opis:

Metoda powoduje ustawienie, jako bieżącej kolejnej pozycjivat dla rejestruvat.

Przykład:

```
Pozycjavat.MoveFirst
do loop
    MsgBox "opis=" + Cstr(Pozycjavat.Value("opis"))
    Pozycjavat.MoveNext
wend
```

3.25.3.8 Insert

Składnia:

```
Function Pozycjavat.Insert() As Boolean
```

Opis:

Metoda dodaje nową pozycjevat dla rejestruvat.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli została utworzona kolejna pozycjavat.

Przykład:

```
If Not rejestruvat.Pozycjavat.Insert() then
    MsgBox "nowa Pozycjavat nie może być utworzona"
Endif
```

3.25.3.9 MoveKursVAT

Składnia:

```
Function Pozycjavat.MoveKursVAT() As Boolean
```

Opis:

Metoda przesuwa pozycjevat na opozycję kursu dla podatku VAT.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli została przesunięta pozycja na pozycję kursu dla podatku VAT.

Przykład:

```
If Pozycjavat.MoveKursVAT() then
    MsgBox "Kurs VAT=" + Cstr(Pozycjavat.Value("wartosc"))
Endif
```

3.25.3.10 MoveKursCITPIT

Składnia:

```
Function Pozycjavat.MoveKursCITPIT() As Boolean
```

Opis:

Metoda przesuwa pozycjevat na opozycję kursu dla podatku CIT/PIT.

Wartość zwracana:

Metoda zwraca wartość TRUE, jeśli została przesunięta pozycja na pozycję kursu dla podatku CIT/PIT.

Przykład:

```
If Pozycjavat.MoveKursCITPIT() then
    MsgBox "Kurs CIT/PIT=" + Cstr(Pozycjavat.Value("wartosc"))
Endif
```

3.25.4 Zdarzenia

Obiekt nie generuje zdarzeń

Przykład:

```
Dim itg As BtDatabase
Dim dok As PDokument
Dim pozVAT As PPozycjaVAT
Dim rej As RejestrVAT
Set itg = New BtDatabase
Set dok = New PDokument
itg.Open "Driver=SQL Server;Server=serversql;Database=bazadanych;", "us", "pass"
itg.CurrentDate = "2015-10-03"
dok.Open "0", itg

...

dok.RejestrVAT.Insert
Set pozVAT = dok.RejestrVAT.PozycjaVAT
pozVAT.MoveFirst
pozVAT.Insert
pozVAT.Value("stawka") = 0
pozVAT.Value("stawka_pl") = 23
pozVAT.Value("wartosc") = 11.11
pozVAT.Value("ue") = 1
pozVAT.Value("usluga") = 1
pozVAT.Value("nazwa") = "jakaś usługa"
pozVAT.Insert
pozVAT.Value("stawka") = 23
pozVAT.Value("stawka_pl") = 23
pozVAT.Value("wartosc") = 33
pozVAT.Value("ue") = 0
pozVAT.Value("usluga") = 1
pozVAT.Value("nazwa") = "jakaś usługa 2"

.MoveKursVAT()
.MoveKursCITPIT()

Set rej = pozVAT.Parent
rej.Value("stawka") = 0
rej.Value("netto") = 300
rej.Value("brutto") = 300
rej.Value("ue") = 1
rej.Value("usluga") = 1
rej.Value("def") = 6

...

dok.insert
```

4 Sposób użycia obiektu dokumentu

4.1 Kolejność operacji dla wprowadzania dokumentów

Utworzenie obiektu BtDatabase i otwarcie połączenia do bazy danych

Utworzenie obiektu PDokument i połączenie dokumentu z bazą danych

Wypełnienie pól dokumentu

Wypełnienie rejestrów VAT

Wypełnienie rozrachunków związanych z dokumentem (dla Faktur i RU)

Utworzenie grupy zapisów

Utworzenie zapisów w grupie

Wypełnienie rozrachunków związanych z zapisem

Sprawdzenie poprawności dokumentu

Odbiór informacji o błędach (pokazanie w UI gdy są)

Zapisanie dokumentu do bazy danych