

XL012 – Wydruki tekstowe

Spis treści

Ogólna struktura mechanizmu wydruków tekstowych

- [Wydruki XML](#)
- [Wydruk tekstowy](#)

Struktura plików XML dla wydruków tekstowych

- [Zawartość tagów](#)
- [Parametry systemowe a typ drukarki](#)

Przykład definicji wydruku

- [Projektowanie wydruku](#)
- [Sekcja SQL](#)
- [Sekcja XSL](#)
- [Sekcja FILE](#)
- [Sekcja RUN](#)

Wymagania systemowe

Drukowanie tekstowe w usługach terminalowych

Ogólna struktura mechanizmu wydruków tekstowych

Wydruki tekstowe zostały oparte o mechanizm wydruków XML, dlatego na początku w skrócie zostanie nakreślona zasada działania wydruków XML.

Wydruki XML

Definicja wydruku XML zawiera 4 sekcje: [SQL], [XSL], [FILE] i [RUN]

Sekcja SQL

Sekcja ta zawiera zapytanie (lub zapytania SQL), które zostaną zadane do bazy danych. Na podstawie zwróconych recordsetów tworzony jest plik XML o strukturze:

```
<ROOT>
```

```
<RECORDSET1>
```

```
<RECORD      NazwaKolumny1="wartość      kolumny      1"  
NazwaKolumny2="wartość kolumny 2" .../>
```

```
<RECORD      NazwaKolumny1="wartość      kolumny      1"  
NazwaKolumny2="wartość kolumny 2" .../>
```

```
</RECORDSET1>
```

```
<RECORDSET2>
```

```
<RECORD      NazwaKolumny1="wartość      kolumny      1"  
NazwaKolumny2="wartość kolumny 2" .../>
```

```
<RECORD      NazwaKolumny1="wartość      kolumny      1"  
NazwaKolumny2="wartość kolumny 2" .../>
```

```
</RECORDSET2>
```

</ROOT>

Tagów RECORDSETn jest tyle, ile recordsetów zwracają zapytania z sekcji [SQL] (są one kolejno numerowane).

Tagów RECORD w każdym tagu RECORDSETn jest tyle, ile wierszy zwróciło dane zapytanie.

Każdy tag RECORD posiada tyle atrybutów, ile kolumn zwraca zapytanie. Nazwa danego atrybutu odpowiada nazwie kolumny, a wartość atrybutu – wartości kolumny, np.

<RECORD Knt_GIDNumer="1" Knt_Kod="COMARCH" />

Sekcja XSL

Sekcja *XSL* zawiera schemat, który jest „nakładany” na plik XML wygenerowany zgodnie z opisem w poprzednim punkcie. W wyniku tego „nałożenia” powstaje plik wynikowy.

Sekcja FILE

Sekcja *FILE* zawiera nazwę pliku, pod jaką zostanie zapisany wynikowy plik XML. Jeżeli sekcja *FILE* jest pusta, wówczas plik wynikowy zostanie zapisany w katalogu tymczasowym pod nazwą rptNNN.tmp (gdzie NNN jest liczbą szesnastkową)

Sekcja RUN

Sekcja *RUN* zawiera nazwę programu, do którego zostanie przesłany plik wynikowy. Jeżeli sekcja *RUN* jest pusta, wówczas plik wynikowy zostanie wysłany do programu domyślnego ze względu na rozszerzenie pliku, np.:

[FILE]

C:\wynik.xml

[RUN]

spowoduje zapisanie pliku wynikowego do pliku *C:\wynik.xml* i otwarcie go za pomocą przeglądarki internetowej.

Parametry dynamiczne

Cała definicja wydruku XML jest poddawana analizie przez silnik parametrów dynamicznych. Może to być wykorzystywane w dwóch przypadkach:

- do przekazywania parametrów dynamicznych wydruku (np. filtr aplikacji, opis wydruku itp.),
- do wywoływania okienka z parametrami, które do wydruku przekazuje użytkownik.

Aplikacja może przekazać do wydruku listę parametrów dynamicznych, zgodnie z zawartością zakładki *Parametry* na oknie *Konfiguracja wydruków*.

Dodatkowo użytkownik może definiować własne parametry dynamiczne, dla których, w chwili wykonywania wydruku, zostanie wyświetlone okno z zapytaniem o wartość parametru.

Opis działania parametrów dynamicznych znajduje się w biuletynie poświęconym modułowi analiz (tworzenie wykresów)

Działanie parametrów dynamicznych zostanie w dalszej części zilustrowane na przykładzie.

Wydruk tekstowy

Do potrzeb wydruków tekstowych przygotowany został program *PrintTxt.EXE*, który jako parametr wejściowy otrzymuje plik XML o określonej strukturze i w wyniku swego działania przetwarza ten XML na plik tekstowy, który jest kierowany na drukarkę.

Pełny mechanizm generowania wydruku polega na przygotowaniu pliku XML o określonej strukturze, a następnie przetworzenie go przez program *PrintTxt.EXE*

Struktura plików XML dla wydruków tekstowych

Plik XML, który jest przekazywany do programu *PrintTxt.EXE* powinien mieć następującą strukturę:

```
<ROOT> <!-- one, req -->
<RAPORT> <!-- many, req -->
<PARAM/> <!-- many, opt -->
<RPT_HEADER> <!-- one, opt -->
<LINE> <!-- many, opt -->
<PARAM/> <!-- many, opt -->
<FIELD/> <!-- many, opt -->
</LINE>
</RPT_HEADER>
<PG_HEADER> <!-- one, opt -->
<LINE> <!-- many, opt -->
<PARAM/> <!-- many, opt -->
<FIELD/> <!-- many, opt -->
</LINE>
</PG_HEADER>
<DETAIL> <!-- many, req -->
<LINE> <!-- many, req -->
<PARAM/> <!-- many, opt -->
```

```
<FIELD/> <!-- many, opt ->

</LINE>

</DETAIL>

<PG_FOOTER> <!-- one, opt ->

<LINE> <!-- many, opt ->

<PARAM/> <!-- many, opt ->

<FIELD/> <!-- many, opt ->

</LINE>

</PG_FOOTER>

<RPT_FOOTER> <!-- one, opt ->

<LINE> <!-- many, opt ->

<PARAM/> <!-- many, opt ->

<FIELD/> <!-- many, opt ->

</LINE>

</RPT_FOOTER>

</RAPORT>

</ROOT>
```

Uwaga

Oczywiście komentarze nie muszą występować w pliku XML – tutaj zostały dodane, aby oznaczyć czy dany tag ma występować, czy też nie:

many oznacza, że tag może wystąpić wiele razy,

one oznacza, że dany tag (na określonym poziomie) może

wystąpić tylko raz,

req oznacza, że dany tag jest wymagany,

opt oznacza, że dany tag jest opcjonalny.

Zawartość tagów

ROOT

Na najwyższym poziomie pliku XML musi wystąpić tylko jeden tag o nazwie *ROOT*

RAPORT

RAPORT zawiera definicję pojedynczego wydruku. Na ogół ROOT zawierał będzie jeden tag RAPORT, jednakże w przypadku konstruowania wydruków seryjnych możliwe jest umieszczenie wielu tagów RAPORT. Istotne jest to, że dla każdego tagu RAPORT zerowana jest numeracja stron.

RPT_HEADER

Tag ten zawiera definicję wyglądu nagłówka raportu. Nagłówek ten zostanie wydrukowany jeden raz na początku raportu.

PG_HEADER

Tag ten zawiera definicję wyglądu nagłówka strony. Nagłówek ten zostanie wydrukowany na początku każdej strony.

DETAIL

Tag ten zawiera definicję zawartości wierszy raportu.

PG_FOOTER

Tag ten zawiera definicję wyglądu stopki strony. Stopka ta zostanie wydrukowana na końcu każdej strony.

RPT_FOOTER

Tag ten zawiera definicję wyglądu stopki raportu. Stopka ta zostanie wydrukowana jeden raz na końcu raportu.

PARAM | P

Tag ten zawiera definicję parametrów. Parametry są zmiennymi raportu, w których można zapisywać pewne informacje (np. takie, do których się potem wielokrotnie odwołujemy) czy dokonywać pewnych obliczeń (np. sumy). Każdy raport posiada pewną predefiniowaną listę parametrów systemowych (takich jak np. numer strony, numer linii, typ drukarki itp.). Nazwy parametrów systemowych są poprzedzone znakiem @.

Jeżeli odwołujemy się do jakiegoś parametru (np. przy obliczeniach) to jego nazwę poprzedzamy znakiem @, stąd odwołanie do parametrów systemowych poprzedzone jest poprzez @@ (np. @@PAGENO)

Tag *PARAM* może posiadać następujące atrybuty:

name	nazwa parametru
type	<p>typ parametru Dopuszczalne wartości to text i number. Domyślnym typem jest text</p> <p>Uwaga: typ parametru jest istotny ze względu na operacje dokonywane na nich.</p> <p>Przykładowo, mając dwa parametry P1 i P2 o wartościach odpowiednio 1 i 2, jeżeli</p> <p>wykonamy na nich operację @P1+@P2, to w przypadku parametrów typu number</p> <p>w wyniku otrzymamy 3, a jeśli co najmniej jeden z nich będzie typu text, to w wyniku</p> <p>otrzymamy 12</p>
value	wartość parametru
calc	<p>przyjmuje wartości true i false (domyślnie false). Określa, czy wartość parametru jest podana przez wartość (calc="true"), czy przez wyrażenie (calc="false"). Przykładowo, jeśli value="@P1+@P2" calc="true" to jako wartość parametru zostanie podstawiony wynik wykonania działania na parametrach P1 i P2. Jeżeli natomiast calc="false", wówczas jako wartość parametru zostałby podstawiony po prostu tekst @P1+@P2.</p>

Uwaga

wszystkie wyrażenia należy tworzyć zgodnie z syntaktyką języka **JScript**. Dotyczy to zmiennych PARAM, pól FIELD jak i obliczania wyrażeń typu true/false (np. dla atrybutów hide). Tagi *PARAM* mogą być elementami tagu *RAPORT* (zainicjalizowanie lub jednorazowe ustawienie parametru na początku wydruku). W tym miejscu można ustawiać niektóre parametry systemowe (np. typ drukarki, nazwa urządzenia wyjściowego) oraz zapamiętywać

w zmiennych stałe wartości parametrów, które będą później wielokrotnie wykorzystywane w raporcie.

Tagi *PARAM* mogą być również elementami tagów *LINE* . wówczas są one przeliczane dla każdej linii.

Dla wydruku definiowane są następujące parametry systemowe

@PRINTER	nazwa urządzenia wyjściowego (np. LPT1, \\KOMPUTER\DRUKARKA, C:\PLIK.TXT – jeżeli chcemy w wydruku zaszyć urządzenie wyjściowe). Możemy również zastosować drukowanie na drukarkę Windows służy do tego parametr dynamiczny ?@Hs100_NQ Printer:''@?.
@VIEWER	nazwa programu służącego do podglądu wydruku, domyślnie jest to „notepad.exe” .
@CODEPAGE	sposób kodowania polskich znaków na drukarce. Aktualnie dostępne wartości MAZOVIA, LATIN i EMULATOR (oznacza tworzenie polskiej litery poprzez wydrukowanie litery łacińskiej, cofnięcie karetki i wydrukowanie ogonka lub akcentu). Ustawienie tego parametru inicjalizuje wartość parametrów @INPUTCODE i @OUTPUTCODE. Domyślnie wartość pusta, oznaczająca brak przekodowania. Zagadnienie dotyczące przekodowania polskich liter omówione jest szczegółowo dalej.
@INPUTCODE	tekst wejściowy dla algorytmu przekodowania polskich liter. Inicjalizowany zgodnie z parametrem @CODEPAGE.
@OUTPUTCODE	tekst wyjściowy dla algorytmu przekodowania polskich liter. Inicjalizowany zgodnie z parametrem @CODEPAGE.

@PRINTER	<p>nazwa urządzenia wyjściowego (np. LPT1, \\KOMPUTER\DRUKARKA, C:\PLIK.TXT – jeżeli chcemy w wydruku zaszyć urządzenie wyjściowe). Możemy również zastosować drukowanie na drukarkę Windows służy do tego parametr dynamiczny ?@Hs100_NQ Printer:''@?.</p>
@PRINTERTYPE	<p>typ drukarki. Aktualnie dostępne wartości to EPSON i PCL. Powodują ustalenie kodów sterujących dla drukarki (w szczególności tryby zagęszczone i pogrubione)</p>
@FONT_NARROW	<p>kod przełączenia drukarki w tryb czcionki zagęszczonej. Inicjalizowany zgodnie z ustawieniem parametru @PRINTERTYPE</p>
@FONT_NARROW_	<p>kod końca tryby czcionki zagęszczonej. Inicjalizowany zgodnie z ustawieniem parametru @PRINTERTYPE</p>
@FONT_BOLD	<p>kod przełączenia drukarki w tryb czcionki pogrubionej. Inicjalizowany zgodnie z ustawieniem parametru @PRINTERTYPE</p>
@FONT_BOLD_	<p>kod końca trybu czcionki pogrubionej. Inicjalizowany zgodnie z ustawieniem parametru @PRINTERTYPE</p>
@CR	<p>znak cofnięcia karetki do początku strony (domyślnie znak o kodzie 13)</p>
@LF	<p>znak przejścia do nowej linii (domyślnie znak o kodzie 10)</p>
@FF	<p>znak końca strony (domyślnie znak o kodzie 12)</p>
@BS	<p>znak cofnięcia karetki o jeden znak (domyślnie znak o kodzie 8)</p>
@PAGENO	<p>numer strony</p>
@PAGETOT	<p>łączna liczba stron wydruku</p>
@LINENO	<p>numer linii na stronie</p>
@PAGELENGTH	<p>liczba wierszy na stronie (domyślnie 60)</p>

LINE | L

Tag ten zawiera definicję wyglądu linii oraz (opcjonalnie) definicje parametrów. Przy przetwarzaniu tagu *LINE* najpierw obliczane są parametry zdefiniowane dla danej linii, a następnie na podstawie zawartości tagów *FIELD* tworzony jest .wygląd. linii na wydruku. Po przetworzeniu tagu *LINE* do pliku wynikowego dodawany jest automatycznie znak nowej linii

FIELD | F

Tag *FIELD* zawiera definicję pojedynczego pola w drukowanej linii. Tag ten może posiadać następujące atrybuty:

name	Jeżeli atrybut nazwa jest ustawiony, to pole jest traktowane tak jak parametr, tzn. wartość aktualnie drukowanego pola jest zapamiętywana jako zmienna
type	Typ pola. Dopuszczalne wartości to text i number (domyślnie text). Od ustawienia typu pola zależy jego formatowanie, wyrównywanie itp.
value	Wartość pola. Może być podana jako wartość (jeżeli calc="false") lub jako wyrażenie (jeżeli calc="true")
align	Sposób wyrównania pól. Dopuszczalne wartości to left, right i center (domyślnie: brak wyrównania dla tekstu i wyrównanie do prawej dla liczb)
width	Szerokość pola.
prec	Dla pól typu number oznacza liczbę miejsc po przecinku.
calc	Określa, czy value zawiera wartość (calc="false") czy wyrażenie (calc="true")
fill	Znak dopełniający pole do pełnej szerokości podanej w width. Domyślnie spacja.
hide	Jeżeli hide="true" to pole jest ukrywane

name	Jeżeli atrybut nazwa jest ustawiony, to pole jest traktowane tak jak parametr, tzn. wartość aktualnie drukowanego pola jest zapamiętywana jako zmienna
hidezero	Jeżeli <code>hidezero="true"</code> to dla pól typu <code>number</code> ukrywane są nieznaczące miejsca po przecinku (w ich miejsce pojawia się spacja)
blankzero	Jeżeli <code>blankzero="true"</code> to pola typu <code>number</code> o wartości 0 są ukrywane.

Uwaga

Wszędzie tam gdzie parametr może przyjmować wartości *true* lub *false* możliwe jest również wprowadzanie wyrażeń. Poniższy przykład ilustruje sposób wydrukowania pola tylko na ostatniej stronie:

```
<LINE>
```

```
<FIELD value="napis na stronie" hide="@@PAGENO!=@@PAGETOT" />
```

```
</LINE>
```

Parametry systemowe a typ drukarki

W poniższym rozdziale zostaną omówione kwestie ustalania kodów sterujących dla drukarek oraz przekodowania polskich liter.

Ustawienie typu drukarki

Obecnie program drukujący potrafi obsługiwać 2 standardy drukarek: *EPSON* (drukarki igłowe) oraz *PCL* (większość drukarek atramentowych i laserowych). Ustalenie typu drukarki jest niezbędne ze względu właściwą obsługę kodów sterujących drukarki, w szczególności przełączania pisma na pogrubione i zawężone.

Jeżeli drukarka wymaga innych kodów sterujących, można to osiągnąć poprzez ustawienie parametrów:

- @FONT_NARROW
- @FONT_NARROW_
- @FONT_BOLD
- @FONT_BOLD_

Przykładowo, ustawienie kodów równoważnych dla trybu *EPSON* można uzyskać poprzez wpisanie:

```
<PARAM calc="true" type="text" name="@FONT_NARROW"
value="String.fromCharCode(15)"/>
```

```
<PARAM calc="true" type="text" name="@FONT_NARROW_"
value="String.fromCharCode(18)"/>
```

```
<PARAM calc="true" type="text" name="@FONT_BOLD"
value="String.fromCharCode(14)"/>
```

```
<PARAM calc="true" type="text" name="@FONT_BOLD_"
value="String.fromCharCode(20)"/>
```

Natomiast ustawienie kodów równoważnych dla trybu *PCL* można uzyskać poprzez wpisanie:

```
<PARAM calc="true" type="text" name="@FONT_NARROW"
value="String.fromCharCode(27)+'&k2S'"/>
```

```
<PARAM calc="true" type="text" name="@FONT_NARROW_"
value="String.fromCharCode(27)+'&k0S'"/>
```

```
<PARAM calc="true" type="text" name="@FONT_BOLD"
value="String.fromCharCode(27)+'(s7B'"/>
```

```
<PARAM calc="true" type="text" name="@FONT_BOLD_"
value="String.fromCharCode(27)+'(s0B'"/>
```

Ustawienie standardu polskich liter

Obecnie program drukujący potrafi obsługiwać polskie litery w standardach Mazovia i Latin oraz emulować polskie litery poprzez wydruk litery łacińskiej i dodanie do niej akcentu czy

ogonka.

Jeżeli drukarka obsługuje polskie litery w jakimś innym standardzie, można uzyskać wydruki poprzez ustalenie wartości parametrów:

- `@INPUTCODE` lista polskich znaków w standardzie Windows, oddzielonych znakami |
- `@OUTPUTCODE` lista polskich znaków w standardzie drukarki, oddzielonych znakami |

Przykładowo, aby uzyskać przekodowanie polskich liter na litery łacińskie (wydruk bez polskich liter) należałoby wpisać:

```
<PARAM          type="text"          name="@INPUTCODE"  
value="Ą|ą|Ć|ć|Ę|ę|Ł|ł|Ń|ń|Ó|ó|Ś|ś|Ż|ż|Ż|ż|"/>
```

```
<PARAM          type="text"          name="@OUTPUTCODE"  
value="A|a|C|c|E|e|L|ł|N|n|O|o|S|s|Z|z|Z|z|"/>
```

Przykład definicji wydruku

Zasady budowania definicji wydruku zostaną omówione na przykładzie wydruku Faktury sprzedaży. Przykład ten został wybrany na względu na stosunkowo prostą formę wydruku oraz powszechność w użyciu. Definicja tego wydruku jest w standardowej bazie wydruków programu. W niniejszym biuletynie omówimy tylko ciekawsze fragmenty formatu wydruku.

Projektowanie wydruku

Dla potrzeb sprawnego wykonania wydruku należy wcześniej dokładnie zaplanować jego wygląd. Ze względu na to, że na wydrukach tekstowych wyrównanie kolumn uzyskuje się przez dobór ich szerokości (liczonej w znakach) szczególnie ważne

jest zaplanowanie rozkładu poszczególnych pól na wydruku. Do potrzeb wydruku przyjęto, że na stronie mieści się 80 znaków w trybie normalnym oraz 133 znaki w trybie zagęszczonym.

Sekcja SQL

Sekcja *SQL* zawiera pięć zapytań – tak więc do „pierwotnego” pliku XML zostanie podstawionych pięć recordsetów.

```
Select Frm_Nazwa1, Frm_Nazwa2, Frm_Nazwa3, Frm_KodP+'  
'+Frm_Miasto As Frm_Miasto, Frm_Ulica, 'NIP: '+ Frm_NIP As  
Frm_Nip From CDN.Firma where Frm_GIDNumer=1
```

Powyższe zapytanie wyciąga dane firmy (pieczętkę firmy) z odpowiedniej tabeli.

```
exec CDN.wydrTabelaPłatnościIVat ??_QFiltr
```

Powyższe zapytanie wyciąga tabelkę płatności oraz vat dla danego dokumentu. Jak widzimy w sekcji SQL oprócz zapytania SELECT możemy również wywołać procedurę serwerową – klauzula EXEC (EXECUTE).

Uwaga

Aby wydruk z procedurą serwerową wykonał się poprawnie, rola ERPRaport musi mieć prawa wykonywania tej procedury. Dla naszej przykładowej procedury:

```
CDN.wydrTabelaPłatnościIVat
```

należałoby dodatkowo wykonać poniższe zapytanie:

```
GRANT EXECUTE ON CDN.wydrTabelaPłatnościIVat TO CDNRaport
```

Zastosowanie w sekcji parametrów procedury wyrażenia *??_QFiltr* spowoduje podstawienie w jego miejsce filtra przekazanego z listy (a więc wydruk będzie zawierał dokładnie płatności i tabelę VAT dla wybranego dokumentu). Użycie wywołania

wyrażenia `??_QFilter` było możliwe dzięki temu, że we wcześniejszym zapytaniu (w sekcji WHERE klauzuli SQL) użyliśmy deklaracji tego parametru `?@H200_NQ|Filter:''@?`. Warto zwrócić uwagę, że parametr dynamiczny został wywołany z opcją H, co spowoduje, że użytkownik nie będzie monitowany o podanie jego wartości (parametr zostanie przekazany z aplikacji). Kolejną opcją, na którą warto zwrócić uwagę to opcja NQ w deklaracji parametru i Q (brak N w tej opcji) w wywołaniu tego parametru. Oznacza ona, że przekazany filtr nie będzie otaczany apostrofami (NQ) lub będzie (Q).

Sekcja XSL

Początkowy fragment definicji, tzn.

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

<xsl:output method = "xml" omit-xml-declaration="yes"/>

<xsl:preserve-space elements="*" />

<xsl:template match="/">

<xsl:for-each select="ROOT">

<xsl:value-of          disable-output-escaping="yes"
select="string('&lt;?xml version=&quot;1.0&quot;
encoding=&quot;windows-1250&quot;?&gt;')"/>
```

można w zasadzie uznać za stały. Ustala on sposób przetwarzania „wejściowego” pliku XML oraz ustawia nagłówek pliku wyjściowego. Następnie rozpoczyna się właściwa deklaracja definicji wydruku.

RAPORT

W tagu *RAPORT* jest ustalonych kilka tagów *PARAM* (a więc

zdefiniowanych kilka zmiennych raportu). Ich znaczenie jest następujące:

Poniższe 3 tagi zawierają ustawienia wartości parametrów systemowych, powodujące ustalenie portu wydruku jako *LPT1*, typu drukarki jako *EPSON* i standardu polskich liter na drukarce jako *MAZOVIA*

```
<PARAM type="text" name="@PRINTER" value="LPT1"/>
```

```
<PARAM type="text" name="@PRINTERTYPE" value="EPSON"/>
```

```
<PARAM type="text" name="@CODEPAGE" value="MAZOVIA"/>
```

Kolejny tag:

```
<xsl:apply-templates select="RECORDSET1"/>
```

powoduje wykonanie template dla recordsetu 1. Są z nim związane następujące dalsze fragmenty w definicji:

```
<xsl:template match="RECORD" mode="r1">
```

```
<P type="text" name="FRM_NAZWA1"><xsl:attribute name="value"><xsl:value-of select="@Frm_Nazwa1"/></xsl:attribute></P>
```

```
<P type="text" name="FRM_NAZWA2"><xsl:attribute name="value"><xsl:value-of select="@Frm_Nazwa2"/></xsl:attribute></P>
```

```
<P type="text" name="FRM_NAZWA3"><xsl:attribute name="value"><xsl:value-of select="@Frm_Nazwa3"/></xsl:attribute></P>
```

```
<P type="text" name="FRM_ULICA" ><xsl:attribute name="value"><xsl:value-of select="@Frm_Ulica"/></xsl:attribute></P>
```

```
<P type="text" name="FRM_MIASTO"><xsl:attribute name="value"><xsl:value-of select="@Frm_Miasto">
```

```
/></xsl:attribute></P>
```

```
<P type="text" name="FRM_NIP" ><xsl:attribute  
name="value"><xsl:value-of select="@Frm_Nip"  
</xsl:attribute></P>
```

```
</xsl:template>
```

Cały ten zabieg powoduje podstawienie wartości pól odczytanych z pierwszego recordsetu do odpowiednich parametrów.

Uwaga

W powyższej definicji `xsl:template` należy zwrócić uwagę na to, żeby we fragmencie `<xsl:value-of select="@Frm_Nazwa1" />` `Frm_Nazwa1` to nazwa kolumny z zapytania SQL i jest ona identyczna (ważne duże, małe litery) !

RPT_HEADER

W tagu `RPT_HEADER` znajduje się pieczętka firmy, dane faktury (numer, data wystawienia, data sprzedaży, numer konta bankowego, dane kontrahentów itp.). Informacje te będą drukowane tylko na pierwszej stronie wydruku.

Zawiera on kilkanaście tagów `L [LINE]` . tak więc nagłówek raportu będzie liczył tyleż linii.

Omówimy tylko kilka linii na które należy zwrócić uwagę:

```
<L>
```

```
<F value="+-Sprzedawca:" width="42" fill="-"/>
```

```
<F value="+" width="5"/>
```

```
<F calc="true" value="@@FONT_BOLD"/>
```

```
<F value="Faktura VAT"/>
```

```
<F calc="true" value="@@FONT_BOLD_"/>
```

```
</L>
```

W powyższej linii na wydruku będzie wydrukowane

+ - Sprzedawca: ————— + Faktura VAT

Czyli na 42 znakach (`width="42"`) zostanie wydrukowany tekst „+ - Sprzedawca” (zawartość atrybutu `value`), 30 znaków minus – (minus bo atrybut `fill="-"`, 30 bo $42 - 12$ gdzie 42 to wartość atrybutu `width`, a 12 to długość stringu w wartości atrybutu `value`). Jest to pierwsze pole w linii. Drugie pole to + i cztery spacje, następne pole to pogrubiony napis Faktura VAT (dzięki wstawieniu znaków sterujących zapisanych w parametrze `@FONT_BOLD`). Następne linie będą drukowane fontem normalny, ponieważ został wstawiony znacznik końca fontu pogrubionego (`@FONT_BOLD_`).

```
<L>
```

```
<F value="|" width="2"/>
```

```
<F calc="true" value="@FRM_NAZWA1" width="40"/>
```

```
<F value="|"/>
```

```
</L>
```

W powyższej linii mamy wstawioną nazwę firmy na długości 40 znaków otoczonej znakami |. Nazwa firmy znajduje się w parametrze `@FRM_NAZWA1`, którego wartość brana jest z zapytania SQL omawianego w sekcji SQL.

```
<L hide="!(@NAB_NAZWA2!=" || @ODB_NAZWA2!=")">
```

```
<F calc="true" value="@NAB_NAZWA2" width="60"/>
```

```
<F calc="true" value="@ODB_NAZWA2" width="60"/>
```

```
</L>
```

Powyzsza linia zawiera drugi czlon nazwy kontrahenta glownego (nabywca) oraz docelowego (odbiorca). Warto zwrócic uwage na atrybut `hide`, który steruje wyświetlaniem tej linii. Linia ta

będzie drukowana wtedy, gdy jedna z wartości @NAB_NAZWA2 lub @ODB_NAZWA2 będzie niepusta, czyli jeżeli na odpowiednich adresach kontrahentów będą wypełnione drugie pola w nazwie kontrahenta. Dzięki temu unikniemy „pustych” linii na wydruku.

PG_HEADER

Kolejnym tagiem jest *PG_HEADER*. Zawiera on 3 tagi *L [LINE]*, tak więc nagłówek strony będzie liczył 3 linie, które są nagłówkiem tabelki z pozycjami faktury.

DETAIL

Kolejny tag *DETAIL* zawiera pozycje faktury. Do jego wygenerowania wykorzystany jest template tworzący linię dla każdego wiersza z recordsetu czwartego

```
<DETAIL>
```

```
<xsl:apply-templates select="RECORDSET4"/>
```

```
</DETAIL>
```

i dalej

```
<xsl:template match="RECORDSET4">
```

```
<xsl:apply-templates select="RECORD" mode="r4"/>
```

```
</xsl:template>
```

```
<xsl:template match="RECORD" mode="r4">
```

```
....
```

```
</xsl:template>
```

Każda pozycja na fakturze generuje na wydruku jedną linię. W ramach tej linii obliczane są pewne zmienne oraz drukowane pola.

RPT_FOOTER

RPT_FOOTER zawiera podsumowanie całego wydruku, czyli tabelkę płaćności, tabelkę VAT, kwotę dokumentu itp.

PG_FOOTER

PG_FOOTER zawiera dolne obramowanie tabelki pozycji faktury, stronę wydruku oraz informuje o nazwie programu, z którego powstał wydruk.

Proszę zwrócić uwagę, że linia, która jest obramowaniem tabelki pozycji drukowana na ostatniej stronie wygląda inaczej niż linia drukowana na innych stronach (ze względów estetycznych, na ostatniej stronie, linia ta stanowi równocześnie górne obramowanie tabelki z płaćności i vat). Zostało to uzyskane przez wykorzystanie atrybutu hide.

```
<L><F value="+--  
+-----+-----+-----+-----+-----+-----+-----+-----+  
hide="@@PAGENO==@PAGETOT"/></L>
```

```
<L/>
```

```
<L hide="@@PAGENO==1">
```

```
<F value="Strona:" width="10"/>
```

```
<F calc="true" value="@@PAGENO" width="3" type="number"/>
```

```
<F value="/"/>
```

```
<F calc="true" value="@@PAGETOT" width="3" type="number"/>
```

```
<F value="CDN XL (C) COMARCH S.A." align="right" width="120"/>
```

```
</L>
```

```
<L hide="@@PAGENO!=1"><F value="ERP XL (C) COMARCH S.A."  
align="right" width="133"/></L>
```

Warto zaznaczyć również, że numer strony w stopce strony nie

drukowany jest na pierwszej stronie tylko na każdej następnej (warunek `@@PAGENO==1`). Na pierwszej stronie, drukowany jest w nagłówku raportu `RPT_HEADER`

Sekcja FILE

Sekcja *FILE* nie została wypełniona „pomocniczy” plik XML będzie więc zapisywany do katalogu TEMP.

Jeżeli w sekcji file wpisujemy np. `C:\FS.XML` to zostanie stworzony plik XML wygenerowany poprzez wydruk XML, jest to plik będący bezpośrednio działaniem mechanizmu wydruków XML i jest on poddawany dalszej obróbce przez program *PrintTXT*.

W ramach testów mechanizmu wydruków można taki plik wygenerować i zobaczyć jego strukturę.

Sekcja RUN

Sekcja *RUN* zawiera wywołanie programu *PrintTXT.EXE*, przetwarzającego plik XML na plik tekstowy wysyłany do drukarki.

Wymagania systemowe

Wymagania systemowe dla poprawnego działania systemu ERP XL zostały opisane w dokumencie *Wymagania Minimalne Systemu do aktualnej wersji...*

Dodatkowo dla poprawnego działania wydruków tekstowych konieczne są komponenty:

- Microsoft Windows Script
- Microsoft Windows Script Control

Wersje instalacyjne powyższych komponentów oraz więcej informacji na ich temat można znaleźć w lokalizacji:

<http://msdn.microsoft.com/library/default.asp?url=/downloads/list/webdev.asp>

Drukowanie tekstowe w usługach terminalowych

Program printtxt.exe który obsługuje drukowanie txt nie pozwala na drukowanie gdy program XL pracuje w usługach terminalowych na drukarki zmapowane w sesji terminalowej. Do obsługi drukowania na drukarkach mapujących się w usługach terminalowych wykorzystano drukowania porze kanału wirtualne. Aby móc drukować wydruki txt w takim środowisku należy:

- W konfiguracji Stanowiska w zakładce **Wydruki i wykresy** zaznaczyć parametr *Wydruki tekstowe przez kanał wirtualny*
- Zainstalować na komputerze gdzie podłączone są drukarki i gdzie uruchamia się klienta połączenia terminalowego program: *PrintTxtDVC.exe* który znajduje się w katalogu XL-a

XL014 – Konfiguracja wydruków

kodów kreskowych

Ogólna struktura mechanizmu wydruków

Wydruki kodów kreskowych w programie ERP XL, realizowane są przy pomocy funkcjonalności tak zwanych wydruków XML. Całość instrukcji dotyczących definicji wydruków zawarta jest w pliku o rozszerzeniu XPT, który jest importowany do programu ERP XL. Dla lepszego zobrazowania posłużono się przykładem pliku *Wydruk.xpt* przeznaczonego dla drukarki Datamax.

Struktura pliku XPT

Plik o rozszerzeniu XPT składa się z dwóch części. Pierwszej [SQL] zawierającej zapytania SQL, oraz drugiej [XSL]. Przykładowy plik *Wydruk.xpt* wygląda następująco:

[SQL]

```
select Twr_Kod,Twr_EAN FROM cdn.TwrKarty WHERE  
?@Hs200_NQ|Filtr:"@?
```

[XSL]

```
<xsl:stylesheet  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
version="1.0">
```

```
<xsl:output omit-xml-declaration="yes"/>
```

```
<xsl:preserve-space elements="*" />
```

```
<xsl:template match="/">
```

```
<xsl:for-each select="ROOT">
```

```

<xsl:apply-templates select="RECORDSET1"/>

</xsl:for-each>

</xsl:template>

<xsl:template match="RECORDSET1">

<xsl:apply-templates select="RECORD"/>

</xsl:template>

<xsl:template match="RECORD">

&#02;&#10;H07&#10;D11&#10;m&#10;1911008000800041&#10;<xsl:valu
e-of select="@Twr_Kod" />&#10;3a6210000000050&#10;<xsl:value-
of select="@Twr_EAN" />&#10;PCS&#10;E&#10;

</xsl:template>

</xsl:stylesheet>

```

Sekcja SQL

Sekcja ta zawiera zapytania SQL (w przypadku naszego przykładu jedno zapytanie), które zostaną zadane do bazy i na podstawie zwróconych odpowiedzi utworzony zostanie plik XML o określonej strukturze.

Uwaga

Zalecane jest, aby nie korzystać z zapytań typu `select *`. Przy wpisywaniu nazw kolumn należy zwrócić uwagę na wielkość liter gdyż XML jest 'case sensitive'

Zastosowanie w sekcji WHERE wyrażenia `?@s200H_NQ|Filtr:"@?` spowoduje wstawienie w jego miejsce filtru przekazanego z programu. Jest to tak zwany parametr dynamiczny. Struktura pliku XML otrzymanego w wyniku zapytań SQL wygląda następująco:

```
<ROOT>
```

```

<RECORDSET1>

<RECORD      NazwaKolumny1="wartość      kolumny      1"
NazwaKolumny2="Wartość kolumny 2" />

<RECORD      NazwaKolumny1="wartość      kolumny      1"
NazwaKolumny2="Wartość kolumny 2" />

</RECORDSET1>

<RECORDSET2>

<RECORD      NazwaKolumny1="wartość      kolumny      1"
NazwaKolumny2="Wartość kolumny 2" />

<RECORD      NazwaKolumny1="wartość      kolumny      1"
NazwaKolumny2="Wartość kolumny 2" />

</RECORDSET2>

.....

</ROOT>

```

Każdy XML wygenerowany z zapytań z sekcji [SQL] rozpoczyna się od Tagu <ROOT>. Tagi <RECORDSETn> odpowiadają kolejnym zapytaniom select tzn. pierwszy select generuje <RECORDSET1> itd. Każdy wiersz z odpowiedzi przedstawiony jest w formie Tagu <RECORD>. Każdy z nich posiada tyle atrybutów, ile kolumn zwraca zapytanie. Nazwa atrybutu odpowiada nazwie kolumny a wartość atrybutu wartości kolumny.

Plik XML dla naszego przykładu może więc wyglądać następująco:

```

<ROOT>

<RECORDSET1>

<RECORD Twr_Kod="MAG770T" Twr_EAN="8" />

</RECORDSET1>

```

</ROOT>

Sekcja XSL

Sekcja zawiera schemat, który jest „nakładany” na XML z danymi. Tłumaczy plik XML na język poleceń drukarki. W naszym przykładzie dla Tagu <RECORD> zdefiniowany został zestaw instrukcji dla drukarki:

```
&#02;&#10;H07&#10;D11&#10;m&#10;1911008000800041&#10;<xsl:valu  
e-of select="@Twr_Kod" />&#10;3a6210000000050&#10;<xsl:value-  
of select="@Twr_EAN" />&#10;PCS&#10;E&#10;
```

W instrukcjach sterujących drukarką osadzone zostały instrukcje xsl-a, pozwalające na wykorzystanie danych uzyskanych z zapytania SQL.

W przypadku innych typów drukarek należy przeczytać dokumentację producenta w celu zapoznania się ze składnią instrukcji sterujących dla drukarki.

Uwaga

W przypadku niektórych drukarek język instrukcji sterujących drukarką jest dość skomplikowany np. drukarki DYMO. Dlatego w ich przypadku należy posłużyć się zestawem obiektów COM, dostarczanych przez producenta.

Struktura pliku XPT będzie więc różnić się od przedstawionej w przykładzie. Przykładowy plik XPT zawierający instrukcje sterujące dla drukarki DYMO wygląda następująco:

[SQL]

```
select Twr_Kod,Twr_Nazwa,Twr_GrupaPod,Twr_EAN FROM  
cdn.TwrKarty where ?@Hs200_NQ|Filtr:"@?
```

[XSL]

```
<xsl:stylesheet  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
xmlns:ms="urn:schemas-microsoft-com:xslt" xmlns:usr="urn:the-
```

```
xml-files:xslt" version="1.0">
<xsl:output omit-xml-declaration="yes" method="html"/>
<ms:script language="VBScript" implements-prefix="usr">
<![CDATA[
Function PrintEan(Ean)
Dim DymoAddIn, DymoLabel
Set DymoAddIn = CreateObject("DYM0.DymoAddIn")
Set DymoLabel = CreateObject("DYM0.DymoLabels")
DymoAddIn.Open "C:\Program Files\Dymo Label\Label
Files\Barcode Label (30252, 30320).LWL"
DymoLabel.SetField "Barcode",Ean
DymoAddIn.Print 1, TRUE
PrintEan=""
End Function
]]>
</ms:script>
<xsl:output omit-xml-declaration="yes"/>
<xsl:preserve-space elements="*" />
<xsl:template match="/">
<xsl:for-each select="ROOT">
<xsl:apply-templates select="RECORDSET1"/>
</xsl:for-each>
</xsl:template>
```

```
<xsl:template match="RECORDSET1">  
  
<xsl:apply-templates select="RECORD"/>  
  
</xsl:template>  
  
<xsl:template match="RECORD">  
  
<xsl:value-of select="usr:PrintEan(123456)"/>  
  
</xsl:template>  
  
</xsl:stylesheet>
```

W przypadku drukarek DYMO najwygodniej posłużyć się językiem skryptowym. W naszym przykładzie jest to VBscript.

```
<ms:script language="VBScript" implements-prefix="usr">
```

Dzięki temu można łatwo zdefiniować funkcje, w których można wykorzystać obiekty COM, dostarczone przez producenta.

XL021 – Specyfikacja standardu wydruków w CDN XL

Spis treści

[Założenia ogólne](#)

Wydruk typu "Lista"

Wygląd dokumentów handlowych (Faktury)

Założenia ogólne

Wszystkie liczby określające położenie i rozmiar obiektów są podane w jednostkach 1/10 mm. Tworzenie elementów w określonej przez opis kolejności, pozwala otrzymać prawidłowy wygląd elementów.

Obszar wydruku

Przyjęcie poniższych wartości pozwala na uzyskanie poprawnego wydruku na wszystkich drukarkach atramentowych (mają one mniejszy obszar drukowania niż laserowe)

Odstęp	Orientacja strony	Portrait	Landscape
	Górny	7 mm	7 mm
	Lewy	7 mm	20 mm
	Prawy	7 mm	7 mm
	Dolny	20 mm	7 mm

Marginesy ustawiane w wydrukach standardowych:

Odstęp	Orientacja strony	Portrait	Landscape
	Górny	10 mm	10 mm
	Lewy	11 mm	20 mm
	Prawy	11 mm	20 mm
	Dolny	20 mm	10 mm

Zmniejszenie marginesów lewego i prawego pozwala na umożliwienie automatycznego przesunięcia wydruku, tak by nadawał się on do wpięcia do skoroszytu. W trybie landscape można zmniejszyć margines lewy albo prawy do wielkości 7 mm –

wydruk zostanie automatycznie dostosowany do trybu landscape
jaki supportuje drukarka (90 lub 270 stopni)

Standard prezentacji danych

Ogólne założenia co do wyglądu elementów wydruku: • Opisy elementów – czcionka normalnej grubości, Times New Roman CE • Kluczowe elementy (wartości liczbowe, daty itp.) – czcionka pogrubiona, Courier New CE • Opisy kolumn list – w ramce o grubości 10; bez pionowych linii dzielących na kolumny, oddalone o 10 jednostek w pionie od poprzedniego elementu

Wydruk typu „Lista”

Obiekty o numerach 1, 2, 3, 4 umieszczamy w sekcji „PAGE HEADER”

Obiekty 5, 6, 7 umieszczamy w sekcji „GRAND TOTALS”

Obiekt 1. [Ramka pieczętki firmy]

Ramka pieczętki firmy:

BOX(Black, White, 275, 5, No, 920, 0, 10) – „Send to back”

BOX(Black, Black, 275, 5, No, 920, 10, 20) – „Send to back”

Obiekt 2. [Elementy pieczętki firmy]

Elementy pieczętki firmy przekazywane są poprzez parametry:
CDN_FIRMA1, CDN_FIRMA2, CDN_FIRMA3, CDN_FMIASTO, CDN_FULICA

Pieczętkę firmy drukujemy czcionką: TIMES NEW ROMAN; rozmiar = 10, normalna grubość

Pola pieczętki firmy umieszczamy jedno po drugim w odstępie

50 jednostek, począwszy od punktu o współrzędnych (10,20). Wszystkie pola wyrównujemy do współrzędnej X=10. Szerokość wszystkich elementów ustawiamy na 900 jednostek.

Obiekt 3. [Tytuł wydruku]

Tytuł wydruku jest odpowiednim, polem tekstowym sformatowanym wg poniższego przykładu:

```
STRING(Silver, Times New Roman CE, 14, Black, Bold, No, No, No, 0, Center, 'Plan Kont', No, 930, 940, 10)
```

LANDSCAPE: do współrzędnej X dodajemy 685

Obiekt 4. [Listwa z numerem strony]

```
BOX(Default, Silver, 55, 0, No, 930, 940, 240) – „Send to Back”
```

```
STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Right, 'Strona:', No, 145, 1675, 242)
```

```
Zmienna z numerem strony: PG#(Silver,,Times New Roman CE, 10, Black, Normal, No, No, No, 0, Left, @n3, No, No, 80, 1795, 242)
```

LANDSCAPE: do współrzędnej X dodajemy 685

Obiekt 5. [Listwa z opisem wydruku]

Sekcja kończąca wydruk rozpoczyna się podkreśleniem elementów oraz umieszczoną pod nim listwą z opisem wydruku (operator drukujący, liczba stron, data sporządzenia wydruku). Dane operatora drukującego są przekazywane do wydruku poprzez zmienną `CDN_OPERATOR`.

Współrzędna Y wszystkich elementów obiektu są zależne od wcześniej umieszczonych elementów zmiennych wydruku (np.

Podsumowania) – dlatego w opisie właściwości obiektów są one zastąpione napisem „YYY”. Na

tej współrzędnej położone jest podkreślenie, współrzędne Y pozostałych obiektów podane są jako przesunięcie w stosunku do położenia podkreślenia (np. YYY+8) – dotyczy to także elementów składających się na obiekt nr.6

HLINE(Black, 10, 1873, 0, YYY)

BOX(Default, Silver, 55, 0, No, 1873, 0, YYY+5) – „Send to back”

STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, ‘Operator drukujący:’, No, 290, 4, YYY+8)

FIELD(Silver, CDN_Operator, Times New Roman CE, 10, Black, Bold, No, No, No, 0, Default, @s50, No, No, 660, 300, YYY+8)

STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, ‘Liczba stron:’, No, 185, 1600, YYY+8)

PG#(Silver,,Times New Roman CE, 10, Black, Normal, No, No, No, 0, Left, @n3, No, No, 80, 1790, YYY+8)

Jeżeli data wydruku nie została umieszczona w sekcji „PAGE HEADER”, to umieszczamy ją na listwie z opisem wydruku, wg poniższych danych: Datę wydruku obliczamy wyrażeniem „TODAY()” – dla jasności opisu formułę zawierającą to wyrażenie nazwę „DZISIAJ”

STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, ‘Data wydruku:’, No, 215, 970, YYY+8)

FIELD(Silver, DZISIAJ, Times New Roman CE, 10, Black, Bold, No, No, No, 0, Left, @d10, No, No, 230, 1190, YYY+8)

LANDSCAPE: Dla obiektów o pełnej szerokości strony (HLINE i BOX) zwiększamy szerokość o 685;

Obiekt ‘Liczba stron:’ ustawiamy na wsp. X =2290

Obiekt 'PG#' ustawiamy na wsp. X=2480

Obiekt 'Data wydruku: ' ustawiamy na wsp. X=1780

Obiekt 'DZISIAJ' ustawiamy na wsp. X=2000

Obiekt 6. [Zastosowany Filtr]

Do wydruku przy pomocy parametru CDN_Filtr przekazywany jest filtr jaki został zastosowany w momencie jego uruchomienia. Ponieważ może on być stosunkowo długi, rozkładamy go na 3 części, tworząc następujące formuły:

```
FILTR_P1: IF CDN_FILTR<>' ' THEN 'Zastosowano filtr:  
'&SUB(CDN_FILTR,1,85) ELSE ''
```

```
FILTR_P2: SUB(CDN_FILTR,86,85)
```

```
FILTR_P3: SUB(CDN_FILTR,172,85)
```

Rozmieszczenie formuł na wydruku:

```
FIELD(Default, FILTR_P1, Courier New CE, 8, Black, Normal,  
Yes, No, No, 0, Default, @s115, Yes, No, 1873, 0, YYY +66)
```

```
FIELD(Default, FILTR_P2, Courier New CE, 8, Black, Normal,  
Yes, No, No, 0, Default, @s115, Yes, No, 1873, 0, YYY +103)
```

```
FIELD(Default, FILTR_P2, Courier New CE, 8, Black, Normal,  
Yes, No, No, 0, Default, @s115, Yes, No, 1873, 0, YYY +140)
```

Jeżeli w wydruku stosujemy własne parametry, mające wpływ na filtr to poniżej umieszczamy napis 'GDZIE' oraz kolejno opis tych parametrów (<nazwa> = <wartość>):

```
STRING(Default, Courier New CE, 8, Black, Normal, Yes, No, No,  
0, Default, 'Gdzie: ', 120, 0, YYY+177)
```

Wszystkie opisy zmiennych drukujemy takim samym fontem jak filtr. Opisy te wyrównujemy do lewej do współrzędnej X=130

LANDSCAPE: W tym trybie ustawiamy szerokość pól na 2565 jednostek

Obiekt 7. [Listwa kończąca wydruk]

Współrzędne Y elementów tego obiektu zależą od ilości danych umieszczonych w obiekcie nr. 6. Odległość od ostatniego elementu obiektu 6 ustalmy na 10 jednostek. Dla czytelności zapisu zamiast wsp. Y wpisany będzie napis „YYY”

BOX(Default, Silver, 55, 0, No, 1873, 0, YYY)

STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, ‘*** Koniec Wydruku ***’, No, 290, 4, YYY+3)

Dodajemy jeszcze pole identyfikujące aplikacje drukującą. Wstawiamy tutaj tekst wg zasady: jeżeli jednoznacznie możemy określić z jakiego modułu pochodzi wydruk, drukujemy napis „CDN XL” + nazwa modułu (np. „CDN XL Księgowość”); W przeciwnym wypadku drukujemy sam napis „CDN XL”

STRING((Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Right, ‘CDN XL ...’, No, 420, 1450, YYY+3)

LANDSCAPE: Elementowi BOX ustawiamy szerokość na 2565 Pole identyfikujące aplikację umieszczamy na wsp. X=2135

Wygląd dokumentów handlowych (Faktury)

Obiekt 1. [Ramka pieczętki firmy]

GROUP(White, Yes, Times New Roman CE, 10, Black, Normal, No, No, No, 330, ‘Sprzedawca’, 920, 0,0) – „Send to back”

BOX(White, White, 310, 0, No, 920, 0,20) – „Send to back”

BOX(Black, Black, 310, 0, No, 920, 13,35) – „Send to back”

Obiekt 2. [Elementy pieczętki firmy]

Elementy pieczętki firmy przekazywane są poprzez parametry: CDN_FIRMA1, CDN_FIRMA2, CDN_FIRMA3, CDN_FMIASTO, CDN_FULICA, CDN_FNIP

Pieczętkę firmy drukujemy czcionką: TIMES NEW ROMAN; rozmiar = 10, Bold

Pola pieczętki firmy (oprócz CDN_FNIP) umieszczamy jedno po drugim w odstępie 45 jednostek, począwszy od punktu o współrzędnych (10,40). Wszystkie pola wyrównujemy do współrzędnej X=10. Szerokość wszystkich elementów ustawiamy na 900 jednostek.

Numer NIP:

STRING(Default, Times New Roman CE, 10, Black, Bold, No, No, No, 0, Default, 'NIP:', 85, 10, 265)

FIELD(Default, CDN_FNIP, Times New Roman CE, 10, Black, Bold, No, No, No, 0, Default, @s13, No, No, 365, 95, 265)

Obiekt 3. [Nagłówek dokumentu]

Tytuł dokumentu: W tytule dokumentu umieszczamy formułę (nazwijmy ją „DOK_TYTUL”), która zwraca nam concatenację typu dokumentu (Faktura VAT, Rachunek Uproszczony itp.) i odpowiednio sformatowanego numeru dokumentu (np. „Faktura VAT nr. 1245/99/CDX”).

FIELD(Silver, DOK_TYTUL, Times New Roman CE, 14, Black, Bold, 0, Center, @s40, No, No, 925, 940,10)

Listwa z numerem strony i oznaczeniem kopii:

BOX(Default, Silver, 65, 0, No, 925, 240, 280) – „Send to back”

STRING(Silver, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Right, 'Strona', 105, 1665, 288)

Zmienna z numerem strony: PG#(Silver,,Times New Roman CE, 10, Black, Normal, No, No, No, 0, Left, @n3, No, No, 80, 1790, 288)

Poprzez parametr CDN_NRKOPII przekazywany jest do wydruku numer drukowanej kopii. Pozwala to na wydrukowanie od razu oryginału i kopii dokumentu. Realizujemy to formułą: ORYGINAL_KOPIA: CHOOSE(CDN_nrKopii>0,' KOPIA',' ORYGINAŁ')

Umieszczoną wg danych: FIELD(Silver, ORYGINAL_KOPIA, Times New Roman CE, 12, Black, Bold, No, No, No, 0, Left, @s10, No, No, 300,940,280)

Zawartość nagłówka oryginału:

STRING(Default, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, 'Data wystawienia:', 280, 1355, 110)

STRING(Default, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, 'Data sprzedaży', 280, 1355, 165)

FIELD(Default, TRN_Data2, Courier New CE, 10, Black, Bold, No, No, No, 0, Right, @d10, No, No, 220, 1645, 110) FIELD(Default, TRN_Data3, Courier New CE, 10, Black, Bold, No, No, No, 0, Right, @d10, No, No, 220, 1645, 165)

Obiekt 4. [Dane nabywcy–odbiorcy / sprzedawcy-dostawcy]

Napis „Nabywca” („Sprzedawca” dla dokumentów zakupu)
STRING(Default, Times New Roman CE, 10, Black, Normal, No, No, No, 0, Default, 'Nabywca:', 220, 10, 350)

Napis „Odbiorca” („Dostawca” dla dokumentów zakupu)
STRING(Default, Times New Roman CE, 10, Black, Normal, No, No,
No, 0, Default, 'Odbiorca', 220, 930, 350)

Adres nabywcy/sprzedawcy (pobrany z tabeli KntAdresy
standardowo połączonej z tabelą TraNag)

FIELD(Default, KnA:Kna_Nazwa1, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s50, No, No, 0, 40, 400)

FIELD(Default, KnA:Kna_Nazwa2, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s50, No, No, 0, 40, 445)

FIELD(Default, KnA:Kna_Nazwa3, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s50, No, No, 0, 40, 490)

Kod pocztowy łączymy z miastem kontrahenta formułą KNTMIAST0:
IF KnA:Kna_KodP='00-000' THEN KnA:Kna_Miasto ELSE
KnA:Kna_KodP&' '& KnA:Kna_Miasto

FIELD(Default, KNTMIAST0, Times New Roman CE, 10, Black, Bold,
No, No, No, 0, Default, @s50, No, No, 0, 40, 535)

FIELD(Default, KnA:Kna_Ulica, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s30, No, No, 0, 40, 580)

NIP kontrahenta drukujemy za pomocą formuły KNTNIP: 'NIP: ' &
KnA:Kna_NipE FIELD(Default, KNTNIP, Times New Roman CE, 10,
Black, Bold, No, No, No, 0, Default, @s30, No, No, 0, 40, 625)

Adres odbiorcy/dostawcy (pobrany z aliasu KntAdresy
połączonego z tabelą TraNag poprzez Trn_AdWGID – GID adresu
wysyłkowego)

FIELD(Default, KnAA:Kna_Nazwa1, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s50, No, No, 0, 950, 400)

FIELD(Default, KnAA:Kna_Nazwa2, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s50, No, No, 0, 950, 445)

FIELD(Default, KnAA:Kna_Nazwa3, Times New Roman CE, 10, Black,

Bold, No, No, No, 0, Default, @s50, No, No, 0, 950, 490)

Kod pocztowy łączymy z miastem kontrahenta formułą ODBMIASTO:
IF KnAA:KnA_KodP='00-000' THEN KnAA:KnA_Miasto ELSE
KnAA:KnA_KodP&' '& KnAA:KnA_Miasto

FIELD(Default, ODBMIASTO, Times New Roman CE, 10, Black, Bold,
No, No, No, 0, Default, @s50, No, No, 0, 950, 535)

FIELD(Default, KnAA:Kna_Ulica, Times New Roman CE, 10, Black,
Bold, No, No, No, 0, Default, @s30, No, No, 0, 950, 580)

Obiekt 5. [Opis dokumentu]

W skład tego obiektu wchodzi opis sposobu dostawy i opis dokumentu (tytuły i wartości). Tytuły tych pól realizujemy formułami: SPOSODBOSTAWY: IF TrN:TrN_SposobDostawy=" THEN " ELSE 'Sposób dostawy: '

OPISDOKUMENTU: IF Tn0:Tn0_Opis=" THEN " ELSE 'Opis dokumentu: '

FIELD(Default, SPOSODBOSTAWY, Times New Roman CE, 9, Black,
Normal, No, No, No, 0, Center, @s16, 300, 10, 685)

FIELD(Default, OPISDOKUMENTU, Times New Roman CE, 9, Black,
Normal, No, No, No, 0, Center, @s16, 300, 10, 730)

FIELD(Default, TrN:TrN_SposobDostawy, Times New Roman CE, 9,
Black, Bold, No, No, No, 0,Default, @s20, 570, 315, 685)

FIELD(Default, Tn0:Tn0_Opis, Times New Roman CE, 9, Black,
Bold, No, No, No, 0,Default, @s255, 1550, 315, 730)

Obiekt 6. [Harmonogram płatności]

Umieszczony jest w sekcji „GRAND TOTALS”

BOX (Default, Silver, 50, 10, No, 918, 0, 0) HLINE(Black, 10,

918,0,50)

STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Forma płatności', Yes, 257, 4, 1)
STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Termin', Yes, 177, 335, 1) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Forma', Yes, 117, 750, 1)

Kolumna form płatności: od współrzędnej (4,54) 4 wiersze co 45 jednostek; picture @s15, szerokość =0 Kolumna terminów płatności: od współrzędnej (335,54) 4 wiersze co 45 jednostek; picture @s10, szerokość =0 Obie powyższe kolumny drukujemy czcionką Times New Roman CE; rozmiar=9 ; normalna grubość

Kolumna kwot płatności: od współrzędnej (561, 54) 4 wiersze co 45 jednostek; szerokość=307. Kwoty drukujemy czcionką Courier New CE; rozmiar=9; Bold

Obiekt 7. [Tabela VAT]

BOX (Default, Silver, 50, 10, No, 930, 940, 0) HLINE (Black, 10, 930,940,50) HLINE (Black, 10, 930,940,105) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Razem:', No, 132, 940, 53) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, ' W tym:', No, 132, 940, 114) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Stawka', No, 130, 1063, 0) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Brutto', No, 108, 1286, 0) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'VAT', No, 82, 1522, 0) STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Center, 'Netto', No, 90, 1760, 0)

Przy dokumencie liczonym „od brutto” kolejność kolumn tabelki VAT ulega zmianie.

Sumy tabelki VAT (nazwy formuł przyjęte dla czytelności opisu)
FIELD(Default, BRUTTOTOT, Courier New CE, 9, Black, Bold, No, No, No, 0, Default, @n-14.2, No, No, 222, 1186, 53)
FIELD(Default, VATTOT, Courier New CE, 9, Black, Bold, No, No, No, 0, Default, @n-14.2, No, No, 196, 1415, 53) FIELD(Default, NETTOTOT, Courier New CE, 9, Black, Bold, No, No, No, 0, Default, @n-14.2, No, No, 246, 1617, 53)

Kolumny tabelki VAT (Courier New CE, rozmiar = 9, normalna grubość) Stawka VAT – od współrzędnych (1081,114) 7 wierszy co 45 jedn. Szerokość=101 Brutto – od współrzędnych (1186,114) 7 wierszy co 45 jedn. Szerokość=222 VAT– od współrzędnych (1415,114) 7 wierszy co 45 jedn. Szerokość=196 Netto – od współrzędnych (1617,114) 7 wierszy co 45 jedn. Szerokość=246

Począwszy od wiersza 4 elementy tabelki VAT mają ustawioną właściwość „Skip if empty” na YES

Obiekt 7. [Stopka strony]

BOX (Default, Silver, 55, 0, No, 1873, 0, 3)

STRING(Silver, Times New Roman CE, 10, black, Normal, No, No, No, 0, Right, 'Razem do zapłaty:', No, 270, 1200, 5)

FIELD(Silver, BRUTTOTOT, Times New Roman CE, 10, black, Bold, , No, No, No, Right, @n-14.2, No, No, 340, 1475, 5)

STRING(Silver, Times New Roman CE, 10, black, Bold, No, No, No, 0, Right, 'zł', No, 50, 1810, 5)

FIELD(Default, KWOTASLOWNIE, Times New Roman CE, 10, black, Normal, No, No, No, 0, Right, @s255, No, No, 1873, 0, 62)

Formuła ZAPLACONO_PRINT drukuje zapłaconą już kwotę faktury i pozostała do zapłaty. Jeżeli nic nie zostało zapłacone to linia jest pusta. FIELD(Default, ZAPLACONO_PRINT, Times New Roman CE, 10, black, Bold, No, No, No, 0, Right, @s255, Yes, No, 1873, 0, 118) HLINE(Black, 10, 1873, 0, 177)

Miejsca na podpisy:

HLINE(Black, 10, 665, 5, 285)

HLINE(Black, 10, 285, 1080, 285)

HLINE(Black, 10, 390, 1470, 285)

STRING(Default, Times New Roman CE, 8, Black, Normal, No, No, No, 0, Default, 'Wystawił:', No, 0, 115, 290) STRING(Default, Times New Roman CE, 8, Black, Normal, No, No, No, 0, Default, 'Data odbioru:', No, 0, 1165, 290) TEXT(No, Default, Times New Roman CE, 8, Black, Normal, No, No, No, 77, Center, Yes, 'Podpis osoby uprawnionej <chr(13),chr(10)> do otrzymania faktury', No, 390, 1470, 290)

Operator wystawiający dokument: FIELD(Default, Ope:Ope_Nazwisko, Times New Roman CE, 8, Black, Bold, No, No, No, 0, Center, @s50, No, No, 450, 226, 290)