

Sisukord

Konfigurowanie wydruków	3
Strojenie	3
Parametry w XSL	5
Przykłady użycia parametrów XSL	6
Kod kreskowy jako obraz	7
Kod GS1	10
Korzystanie z kodu kreskowego Code128 ze skryptem	10
Korzystanie ze skryptu w XSL	10
Ribakoodi funktsioon	11
Funktsiooni väljakutsumine	11
Näidis	11
Pilt tabeli taustaks	14
Lingi kasutamine	14
Signeerimise tugi	14
PDF manuseks	16
EveryPay Linkpay	16
Alustamine	16
Seadistamine	17
HMAC signatuuri loomine välisele lingile	18
Näide	18
Nipid	18
0 (mitte NaN), kui pole andmeid	18
Mingi bloki X korda joonistamine	19
Node-set XML muutuja, sealt unikaalsete kirjete kuvamine	19
dateAdd - kuupäevale mingi päevade arvu lisamine	20
Splitter - Mingi tekstilise sisu laiali jagamine separaatori alusel	21
dateDiff - kuupäevade vahemiku päevade arvu arvutamine	21
Textarea sisu (rea vahedega, näiteks sündmuse kirjeldus) kuvamine ka väljatrükis reavahedega	22
Page break	22
Summeerimine	22
Palgateatisele puhkusejäägi kuvamine	23



Ta strona nie jest jeszcze kompletnie przetłumaczona. Pomoc mile widziana.

(po skompletowaniu usuń ten komentarz)

Konfigurowanie wydruków

Wydruk Directo jest w zasadzie projektem HTML , który jest opisany przy użyciu arkusza stylów XSL. Podczas drukowania dane pobierane są z pliku XML (każdy dokument i raport ma swój własny zbiór plików XML).

Strojenie

Aby skonfigurować wydruki, użytkownik musi mieć uprawnienia, które można ustawić na karcie INNE

Muutmisõigused		
Keelatud	Lubatud	Õigus
<input type="radio"/>	<input checked="" type="radio"/>	Lehitseja muutmine
<input type="radio"/>	<input checked="" type="radio"/>	Kasutajapõhine peenhäälestus
<input type="radio"/>	<input checked="" type="radio"/>	Vormi muutmine
<input type="radio"/>	<input checked="" type="radio"/>	Allsüsteemist tulnud kannete muutmine
<input type="radio"/>	<input checked="" type="radio"/>	Väljatrükkide muutmine

uprawnień użytkownika/grupy.

Seadistused> Üldised seadistused > Väljatrükkide häälestamine

Każdy typ dokumentu może mieć kilka różnych wydruków. Użytkownik może wybrać odpowiedni formularz podczas drukowania lub wysyłania wiadomości e-mail, klikając prawym przyciskiem myszy przycisk **Drukuj** lub **Wyślij e-mailem** . Jeżeli użytkownik nie wybierze wydruku, Directo robi to automatycznie.

W przypadku wyboru automatycznego używany jest pierwszy szablon wydruku spełniający wszystkie opisane warunki, tzn. język, projekt, obiekt itp. określone w ustawieniach odpowiadają wartościom w drukowanym dokumencie. Aby ustalić kolejność automatycznego wyboru, należy zmienić wartość pola Kolejność w ustawieniach – wybór szablonów wydruku będzie przebiegał w kolejności rosnącej zgodnie z tym numerem.

!:Należy zawsze skonfigurować co najmniej jeden szablon wydruku, który nie ma przypisanych żadnych filtrów i jest ostatnim w sekwencji (o najwyższym numerze sekwencji). Zapobiega to sytuacji, w której nie można wybrać żadnego formularza i drukowanie się nie powiedzie.

W przypadku szablonu wydruku, dla którego w kolumnie **Przycisk** wybrano opcję Drukuj, Wyślij e-mailem lub Drukuj+Wyślij e-mailem , odpowiednia akcja pojawi się jako osobny przycisk w nagłówku

dokumentu. Dzięki temu możesz szybko i wygodnie korzystać z różnych szablonów wydruku, bez konieczności wybierania ich prawym przyciskiem myszy.

Przycisk „ **Wyczyść wszystko** ” czyści dodatkowe opcje wydruku. Na przykład informacje o języku, projekcie, serii, obiekcie, magazynie itp.

Nr	Kirjeldus	Keel	Projekt	Seeria	Objekt	Kontaktigrupp	Asukoht	Ladu	Makse-tingimus	Tüüp	Staatus	Valjad	Maatriks	Nupp	Suletud	Järjekord	Värv	Signeeri			
1	Invoice (PDF)	ENG										Vali				5			XML	XSL	Vaata
2	Arve (PDF)											Vali				10			XML	XSL	Vaata
3	Invoice	ENG										Vali				15			XML	XSL	Vaata
4	Arve											Vali				20			XML	XSL	Vaata
5	Invoice (cut)	ENG										Vali				25			XML	XSL	Vaata
6	Arve (lõigatud)											Vali				30			XML	XSL	Vaata
7	Arve Metall											Vali				130			XML	XSL	Vaata
8												Vali							XML	XSL	Vaata
9												Vali							XML	XSL	Vaata
10												Vali							XML	XSL	Vaata
11												Vali							XML	XSL	Vaata
12												Vali							XML	XSL	Vaata

Projekt formularza wydruku znajduje się pod przyciskiem **XSL** . Po prawej stronie okna znajduje się przegląd całego kodu, który ułatwia nawigację i pozwala na szybkie przechodzenie do różnych sekcji lub poruszanie się pomiędzy nimi. W miarę wpisywania kodu edytor podpowiada dopasowania i pomaga w uzupełnianiu kodu. Zmiany muszą zostać zapisane, aby zostały zastosowane.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
3   <xsl:output method="html"/>
4   <xsl:decimal-format name="ocra" decimal-separator='.' grouping-separator=' ' />
5
6   <!--mailto:Arve {nr}-->
7   <!--mailattachment:Arve_{nr}-->
8   <!--pdfparams:HeaderHeight=79,FooterHeight=30,debug_pdf=0,version=1,Encoding=UTF-8-->
9   <!--outxput-pdf-->
10
11   <xsl:template name="header">
12     <center>
13       <table xheight="100" width="800">
14         <tr>
15           <td height="20">&#160;</td>
16         </tr>
17         <tr>
18           <td align="right" valign="top">
19             <xsl:if test="/documents/footer/firma_logo!="">
20               <img border="0" height="60">
21                 <xsl:attribute name="SRC">
22                   <xsl:value-of select="/documents/footer/firma_logo"/>
```

Wszystkie zapisy są przechwytywane w linku **XSL** w każdym wierszu , dzięki czemu możliwe jest otwarcie wcześniejszej wersji. Różnice w kodach można również zobaczyć korzystając z **linków Różnica od ostatniego** lub **Różnica od bieżącego** i **Różnica od następnego** .



Parametry w XSL

Jeśli konieczna jest zmiana ustawień domyślnych, w wydruku można użyć następujących słów kluczowych:

- `<!-- mailfrom:adres_email_nadawcy -->`
- `<!-- mailname:nazwa_nadawcy -->`
- `<!-- mailto:do kogo wysłać -->`
- `<!-- mailnopdf:1 -->` zamiast pdf zostanie utworzony html
- `<!-- mailnoattachment: -->` nie zostanie dodany żaden załącznik PDF, tylko napisany przez Ciebie tekst
- Parametry PDF
 Parametry są rozdzielone przecinkami. Przykład: Plik PDF jest tworzony w formacie poziomym:
`<!-- pdfparams:BrowserWidth=1000,isLandscape=1,version=1,HeaderHeight=60,FooterHeight=60 -->`

ParameterExplanation **debug_pdf=1** Wizualna widoczność strefy nagłówka i stopki
BrowserWidth=800 Szerokość strony podczas rysowania w pikselach (domyślnie 800, 1000 dla orientacji poziomej) **isLandscape=1** Plik PDF jest tworzony w formacie poziomym (szerokość większa od wysokości) **version=1** Wymagany parametr, dzięki któremu współrzędne są obliczane od krawędzi strony do krawędzi piksela **Silnik=1** Renderer Gecko (zamiast IE) UWAGA! układy „płynące” na stronie nie działają z tym **HeaderHeight=60** Wysokość nagłówka mm **FooterHeight=60** Wysokość stopki mm **HeaderHeight1=60** Wysokość nagłówka strony głównej mm (jeśli ma być inna niż pozostałe) **FooterHeight1=60** Wysokość stopki strony głównej mm (jeśli ma być inna niż pozostałe) **FontEmbed=1** dodaje czcionki PDF (można dodawać z [installitud fontide](#) na serwerze) **InsetX=10,InsetY=10** Współrzędne X i Y dla początku rysowania **BrowserWidth=800** umożliwia ustawienie tzw. poziomu Zoom, czyli jak duży jest widok (domyślnie 800) **ImageQuality=10** kompresja pliku

graficznego 0 - 100 (0=skompresowany, słaba jakość, 100=nieskompresowany, bez utraty jakości) **MediaBox=** rozmiar strony A4 (domyślnie A4) **Rect=0 0 210 297** media wymiary „kwadratu” mm, przykładowo A4 (0 0 = x i y dla punktu początkowego rysowania w lewym dolnym rogu strony; 210 - szerokość mm; 297 - wysokość mm)

- <!-- output-excel --> dodaje nagłówek programu Excel - plik jest otwierany za pomocą programu Excel
- <!-- clear-output --> wynik jest tylko czystą transformacją XSL (bez nagłówka Directo)
- Podczas wysyłania wiadomości e-mail tworzone jest zdarzenie, jeśli zachodzi taka potrzeba, parametry tworzonego zdarzenia można zmienić za pomocą odpowiednich parametrów XSL <!-- ev_type:xxx --> <!--ev_status:xxx --> <!--ev_object:xxx --> <!--ev_start:xxx --> <!--ev_end:xxx -->
- <!--output-pdf--> wydruk otwiera się zawsze w formacie PDF (w przypadku raportów warto ustawić także parametr mailattachment określający nazwę pliku - patrz kolejny punkt)
- <!-- mailattachment:attachment/file_name --> może używać znacznika {nr} i parametrów {param:param_name}
- <!-- mailsubject:new_subject --> może używać znacznika {nr} i parametrów {param:param_name}
- <!-- plik-wyjściowy:nazwa-pliku.txt --> plik wyjściowy
- <!-- charset:windows-xxx --> Zwykle kodowanie *pliku wyjściowego* to windows-1257, chyba że w pliku xsl określono *charset=utf-8* (wtedy kodowanie to oczywiście UTF-8). Jeśli jednak parametr charset zostanie określony osobno, użyte zostanie kodowanie określone w tym miejscu.
- <!-- mailbody:xxx --> W przypadku wiadomości e-mail masowej, treść wiadomości. Jeśli xxx jest liczbą, treść jest pobierana z tekstu zawierającego podaną liczbę, jak opisano `Üldised seadistused` Tekstidponizej.

Dla parametrów **mailsubject**, **mailattachment**, **output-file**, **mailfrom**, **mailto** i **mailname** możesz użyć:

- dodatkowe parametry {param:param_name} :!są używane jako nazwy: time1, time2, project, object, language, client_code, warehouse, condition, nazwa może się nieznacznie różnić w zależności od dokumentu (client_code vs. kl_code), zgodnie z definicją w XML
- numer dokumentu {nr} i data miesiąc rok dzień odpowiednio {mm} {yy} {dd}
- niektóre pole danych z XML {param:xml:/documents/contact/class}

Przykłady użycia parametrów XSL

- <!-- mailsubject:DobryRaport od {param:time1} do {param:time2} -->
- <!-- mailsubject: zaproszenie do składania ofert {no} Do obiektu: {param:object} -->
- <!-- mailattachment: myAttachment_number_{nr}_to_class{param:xml:/documents/contact/class} -->

Wskazówka: aby przetestować, możesz ustawić opcję „Edytuj adres e-mail: Tak” na karcie Użytkownika, a następnie otworzy się okno wiadomości e-mail z już utworzonym tematem (aby sprawdzić, czy wszystkie parametry zadziałały).

Kod kreskowy jako obraz

Kod kreskowy można utworzyć w postaci obrazu. Aby to zrobić, należy utworzyć element **img** i wprowadzić adres generatora kodów kreskowych ze szczegółowymi parametrami jako jego wartość **src**. W przykładzie artykuł jest wyprowadzany z pliku XML jako parametr.

```
<img><xsl:attribute name="src">/logos/qr.asp?t=<xsl:value-of
select="artikkel"/>&code=C128&h=30</xsl:attribute></img>
```

Możliwe parametry:

- **t** - dane wejściowe, z których tworzony jest kod kreskowy (w zależności od standardu może to być również tekst)
- **h** to wysokość w pikselach
- **kod** - pożądany standard kodu kreskowego (według poniższej tabeli)

Standard	wartość parametru kodu
Kod QR	qr
KOD 39	C39
KOD 39 SUMA KONTROLNA	C39c
KOD 39E	C39E
KOD SUMY KONTROLNEJ 39E	C39Ec
KOD 93	C93
STANDARD 2 5	S25
STANDARD 2 5 SUMA KONTROLNA	S25c
PRZEPLLOT 2 5	I25
PRZEPLLOT 2 5 SUMA KONTROLNA	I25c
KOD 128	C128
KOD 128A	C128A
KOD 128B	C128B
KOD 128 C	C128C
EAN2	EAN2
EAN5	EAN5
EAN8	EAN8
EAN13	EAN13
UPC-A	UPCA
UPC E	UPCE
MSI	MSI
SUMA KONTROLNA MSI	MSIc
POSTNET	POSTNET
PLANETA	PLANETA
RMS4CC	RMS4CC
KIX	KIX
IMB	IMB
W BARZE	W BARZE
KOD 11	KOD11
KODEKS FARMACEUTYCZNY	FARMACJA

Standard	wartość parametru kodu
KODEKS FARMACYJNY DWA ŚCIEŻKI	FARMACJA2T
W AusPost 4 kod klienta stanowego	usługa pocztowa
Kod Azteków	kod aztecki
Kompaktowy kod Aztec	azteccodecompact
Runy Azteków	runa aztecka
BC412	bc412
Kod kanału	kod kanału
Blok kodowy F	kod blokowy
Kod 11	kod11
Kod 128	kod128
Kod 16K	kod16k
Kod 25	kod2z5
Włoski Pharmakod	kod32
Kod 39	kod39
Kod 39 rozszerzony	kod39ext
Kod 49	kod49
Kod 93	kod93
Kod 93 rozszerzony	kod93ext
Kod Pierwszy	kod
COOP 2 z 5	współpraca2z5
Niestandardowa symbolika 4-stanowa	stuknięty
Rozszerzony GS1 DataBar	rozszerzony pasek danych
Rozszerzony kompozyt GS1 DataBar	rozszerzonykompozyt paska danych
GS1 DataBar rozszerzony ułożony w stos	rozszerzony pasek danych ułożony w stos
GS1 DataBar rozszerzony kompozytowy	pasek danychrozszerzonyułożonykompozyt
GS1 DataBar Ograniczony	ograniczony pasek danych
GS1 DataBar Limited Kompozyt	pasek danychlimitedcomposite
GS1 DataBar wielokierunkowy	pasek danych
GS1 DataBar wielokierunkowy kompozytowy	databaromnicomposite
GS1 DataBar ułożony w stos	ułożony w stos pasek danych
GS1 DataBar Ułożony Kompozyt	pasek danych ułożony w stos kompozytowy
GS1 DataBar układany wielokierunkowo	pasek danychstackedomni
GS1 DataBar Ułożony Wielokierunkowy Kompozyt	pasek danych ułożony w stosy
GS1 DataBar skrócony	pasek danych obcięty
GS1 DataBar skrócony kompozyt	pasek danychobciętykompozyt
Datalogic 2 z 5	danelogic2of5
Macierz danych	macierz danych
Macierz danych prostokątna	macierz danychprostokątny
Kod DotCode	kod kropki
EAN-13	ean13
Kompozyt EAN-13	ean13kompozycja
GS1-14	ean14
EAN-2 (dodatek 2-cyfrowy)	ean2
EAN-5 (dodatek 5-cyfrowy)	5 stycznia
EAN-8	ean8

Standard	wartość parametru kodu
Kompozyt EAN-8	ean8kompozyt
Znaczki Flutter	znaczki trzepoczące
GS1-128	gs1-128
GS1-128 Kompozyt	gs1-128kompozyt
Komponent kompozytowy GS1 2D	gs1-cc
Macierz danych GS1	macierz danych gs1
GS1 Data Matrix Prostokątny	gs1datamatrixprostokątny
Kupon GS1 dla Ameryki Północnej	gs1northamericankupon
Kod QR GS1	kod gs1qr
Kod Han Xin	hanxin
Kod HIBC Aztec	kod hibcaztec
HIBC Codablock F	hibccodablockf
Kod HIBC 128	kod hibccode128
Kod HIBC 39	kod hibccode39
Macierz danych HIBC	macierz danych hibc
Macierz danych HIBC prostokątna	macierz danych hibcprostokątny
HIBC Mikro PDF417	hibcmicropdf417
HIBC PDF417	pdf417
Kod QR HIBC	kod hibcqr
IATA 2 z 5	iata2of5
Niemiecki kod identyfikacyjny pocztowy	kod identyfikacyjny
Przemysłowy 2 z 5	przemysłowy2z5
Przeplatane 2 z 5 (ITF)	przeplatany2z5
Numer ISBN	Numer ISBN
Numer identyfikacyjny	ismn
Numer ISSN-u	numer
ITF-14	tf14
W kodzie klienta Japan Post 4 State	japanpost
Królewski holenderski TPG Post KIX	srać
Kod pocztowy Deutsche Post	kod pocztowy
Macierz 2 z 5	macierz2z5
Kod Maxi	maksykod
MikroPDF417	mikropdf417
Mikrokod QR	Kod QR
MSI zmodyfikował Plessey	msi
Inteligentna poczta USPS	jeden kod
PDF417	pdf417
Kompaktowy PDF417	pdf417kompaktowy
Kod binarny farmaceutyczny	farmakod
Dwutorowy Pharmacode	farmakodek2
Poczta Polska	planeta
Plessey Wielka Brytania	proszę bardzo
Kod pozycji	kod pocztowy
Poczta USPS	sieć pocztowa
Numer Centralnego Rejestru Farmaceutycznego (PZN)	pzn

Standard	wartość parametru kodu
Kod QR	Kod QR
Kodabar	zracjonalizowanyCodabar
Niestandardowa symbolika 1D	surowy
Royal Mail 4 Kod klienta stanowego	poczta królewska
SSCC-18	sscc18
Różne symbole	symbol
W kolonii	w kolonii
Numeryczny w Kolonii	telenumeryczny
Ultrakod	ultrakod
UPC-A	upca
UPC-A Kompozyt	upcacomposite
UPC-E	upce
Kompozyt UPC-E	upcecomposite

Kod GS1

Standard GS1 pozwala na przesyłanie w kodzie danych o kilku różnych parametrach. Na przykład numer seryjny, data przydatności do spożycia itp. Konkretny parametr, czyli tzw. **AI** jest zdefiniowana w standardzie. Podczas generowania kodu, AI powinno być przedstawione w nawiasach, np. numer seryjny kodu kreskowego, a najlepszy do spożycia będzie kod w formacie (01)ribakood(21)seerianr(15)YYMMDD

kodu kreskowego GS1-128 .

[qr.asp](#)

Kod QR GS1

[qr.asp](#)

Korzystanie z kodu kreskowego Code128 ze skryptem

Kod kreskowy zgodny ze standardem Code 128 może być stosowany w wydrukach bez konieczności stosowania zewnętrznych czcionek. Rozwiązaniem jest zaoferowanie wydruku, w którym obraz w formacie HTML jest generowany dla kodu kreskowego.

Należy zwrócić uwagę na następujące kwestie:

Korzystanie ze skryptu w XSL

Podczas deklarowania strony `<xsl:stylesheet>` należy również zdefiniować następujące atrybuty skryptu:

- `xmlns:script = "http://topxml.com/forum/script"`

- `xmlns:x="ignoruj" wyklucz-prefiksy-wyników="skrypt"`

```
<xsl:stylesheet version="1.0" xmlns:str="http://xslt.org/string"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:msxsl="urn:schemas-
microsoft-com:xslt" extension-element-prefixes="str" xmlns:script =
"http://topxml.com/forum/script" xmlns:x="ignore" exclude-result-
prefixes="script">
  <xsl:output method="html"/>
  <xsl:decimal-format name="ocra" decimal-separator='.' grouping-separator='
' />

  <msxsl:script language="JScript" implements-prefix="script">
<![CDATA[
]]>
  </msxsl:script>

</xsl:stylesheet>
```

Ribakoodi funktsioon

Kõik `<msxsl:script>` blokis tuleks ka oma kujundusse kopeerida (siin asub funktsioon, mille abil tehakse ribakoodi pilt)

Funktsiooni väljakutsumine

- Ribakoodi kuvamiseks kutsutakse välja eelpool mainitud funktsioon

```
<xsl:value-of disable-output-escaping="yes"
select="string(script:code128(string(artikkel)))/>
```

- Võib juhtuda, et mõningate sümbolite korrektsena kuvamiseks tuleb ribakoodilugejat seadistada õiget tüüpi klaviatuuri emuleerima (testitud käpaga õnnestus kood lugeda, kui klaviatuuri tüübiks oli määratud SWE/FIN)

Näidis

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt"
  xmlns:script = "http://topxml.com/forum/script"
  xmlns:x="ignore" exclude-result-prefixes="script">
  <xsl:output method="html"/>
  <xsl:decimal-format name="ocra" decimal-separator='.' grouping-
separator=' ' />
```

```

<xsl:template match="/">
  <meta http-equiv="Content-Type" content="text/html;
charset=utf-8"></meta>
  <html>
  <style>
    .bcdiv {float:left;height: 30px; /*size*/}
    .bar1 { border-left:1px solid black; }
    .bar2 { border-left:2px solid black; }
    .bar3 { border-left:3px solid black; }
    .bar4 { border-left:4px solid black; }
    .space0 { margin-right:0 }
    .space1 { margin-right:1px }
    .space2 { margin-right:2px }
    .space3 { margin-right:3px }
    .space4 { margin-right:4px }r
    label {clear:both;display:block;text-align:center; font:
0.125in/100% helvetica;}
  </style>
  <body>
    <table>
      <xsl:for-each select="/documents/document/rows/row">
        <tr>
          <td><xsl:value-of select="artikkel"/></td>
          <td><xsl:value-of disable-output-escaping="yes"
select="string(script:code128(string(artikkel)))"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>

```

```

<msxsl:script language="JScript" implements-prefix="script"><![CDATA[
var BARS
=
[212222,222122,222221,121223,121322,131222,122213,122312,132212,221213,22131
2,231212,112232,122132,122231,113222,123122,123221,223211,221132,221231,2132
12,223112,312131,311222,321122,321221,312212,322112,322211,212123,212321,232
121,111323,131123,131321,112313,132113,132311,211313,231113,231311,112133,11
2331,132131,113123,113321,133121,313121,211331,231131,213113,213311,213131,3
11123,311321,331121,312113,312311,332111,314111,221411,431111,111224,111422,
121124,121421,141122,141221,112214,112412,122114,122411,142112,142211,241211
,221114,413111,241112,134111,111242,121142,121241,114212,124112,124211,41121
2,421112,421211,212141,214121,412121,111143,111341,131141,114113,114311,4111
13,411311,113141,114131,311141,411131,211412,211214,211232,23311120]
, START_BASE = 38
, STOP = 106 //BARS[STOP]==23311120 (manually added a zero at the
end)
;

function code128(code, barcodeType) {
  if (arguments.length<2) barcodeType = code128Detect(code);

```

```
    if (barcodeType=='C' && code.length%2==1) code = '0'+code;
    var a = parseBarcode(code, barcodeType);
    return bar2html(a.join('')) + '<label>' + code + '</label>';
}

function bar2html(s) {
    for(var pos=0, sb=[]; pos<s.length; pos+=2) {
        sb.push('<div class="bcdiv bar' + s.charAt(pos) + ' space' +
s.charAt(pos+1) + '"></div>');
    }
    return sb.join('');
}

function code128Detect(code) {
    if (/^[0-9]+$/.test(code)) return 'C';
    if (/[a-z]/.test(code)) return 'B';
    return 'A';
}

function parseBarcode(barcode, barcodeType) {
    var bars = [];
    bars.add = function(nr) {
        var nrCode = BARS[nr];
        this.check = this.length==0 ? nr : this.check + nr*this.length;
        this.push( nrCode || ("UNDEFINED: "+nr+"->"+nrCode) );
    };
    bars.add(START_BASE + barcodeType.charCodeAt(0));
    for(var i=0; i<barcode.length; i++) {
        var code = barcodeType=='C' ? +barcode.substr(i++, 2) :
barcode.charCodeAt(i);
        converted = fromType[barcodeType](code);
        if (isNaN(converted) || converted<0 || converted>106) throw new
Error("Unrecognized character ("+code+") at position "+i+" in code
'"+barcode+"'");
        bars.add( converted );
    }
    bars.push(BARS[bars.check% 103], BARS[STOP]);
    return bars;
}

var fromType = {
    A: function(charCode) {
        if (charCode>=0 && charCode<32) return charCode+64;
        if (charCode>=32 && charCode<96) return charCode-32;
        return charCode;
    },
    B: function(charCode) {
        if (charCode>=32 && charCode<128) return charCode-32;
        return charCode;
    },
    C: function(charCode) {
        return charCode;
    }
}
```

```
}  
}  
]]></msxsl:script>  
</xsl:stylesheet>
```

Pilt tabeli taustaks

Pildi saab kasutada tabeli taustana, selleks tuleb määrata tabeli stiil järgnevalt:

```
<table style="background: url(/logos/files/pilt.png);background-size: 150px;  
background-position: 260px 10px;background-repeat:no-repeat;" border="0"  
cellpadding="0" cellspacing="0" valign="top">
```

Pdfparams parameeter Engine peab antud juhul olema 1.

```
<!--  
pdfparams:BrowserWidth=1000,isLandscape=1,version=1,HeaderHeight=60,FooterHeight=60,Engine=1-->
```

Lingi kasutamine

```
<a target="_blank">  
  <xsl:attribute name="href">  
    <xsl:value-of disable-output-escaping = "yes"  
select="artikkel_andmed/url"/>  
  </xsl:attribute>  
  <xsl:value-of disable-output-escaping = "yes" select="keelne_seletus"/>  
</a>
```

* Antud näites artikli pealkirjale vajutades suunatakse lingile, mis on seadistatud artiklikaardi väljale URL. Väline link peab olema kujul

```
http://www.directo.ee
```

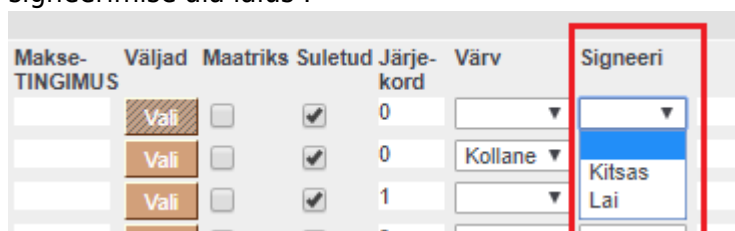
Signeerimise tugi

Signeerimine väljatrüki kontekstis tähendab seda, et dokumendile kuvatakse nupp **Signeeri**. Tavaliselt on see kasutuses olukorras, kus on vaja kliendilt küsida allkirja, näiteks mingi akti allkirjastamine kliendi juuresolekul. Selleks luuaks spetsiaalne väljatrükk signeerimiseks, Signeeri nupu vajutus dokumendil avab selle ning väljatrükil kuvatakse ala allkirjastamiseks, mida saab siis kas arvutist hiirega või tahvlist näpuga „sodides“ allkirjastada. Peale **Valmis** nupu vajutamist salvestub väljatrükk koos joonistatud allkirjaga väljatrükis ettenähtud kohta selle sama dokumendi külge PDF manuseks.

Dokumendid, mis toetavad signeerimist:

- Pakkumine
- Tellimus
- Lähetus
- Arve
- Sündmus
- Klient
- Leping
- Liikumine
- Personal

Signeerimise väljatrüki lisamiseks tuleb väljatrüki definitsioonile määrata tulbast **Signeeri** signeerimise ala laius :



XSL-is tuleb allkirja koht määrata järgmiselt:

```
<img sign="signhere" width="600" border="0" />
```

- Soovikorral kuvatakse eraldi sisestuskast, kuhu saab tekstina sisestada allkirjastaja nime.

Signeeri

Toomas Tava

Toomas Tava

Vali

Kaimar Karu

Toomas Tava

TOOMAS

Selleks tuleb lisada väljatrükki eraldi *div* näites toodud *id*-ga, sisestatud nimi kuvatakse *div*i sisse. Võimalus on ka kuvada eeldefineeritud nimedega rippmenüü (lang atribuut)

```
<div lang="Kaimar Karu,Toomas Tava" id="signer"></div>
```



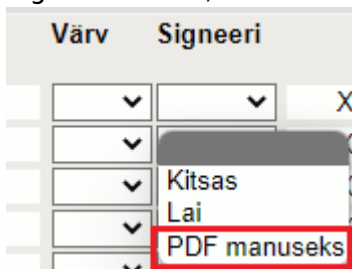
atribuut *id*=„signer“ peab olema *div*-i viimane atribuut.

- võimalus on signeerimist kasutada ka lihtsalt nõ manusena PDFi salvestajana. Sellisel juhul signeerimist ei toimu, lihtsalt signeerimise väljatrükk salvestatakse PDFina dokumendi manuseks. Selle saavutamiseks tuleks lisada XSLi ülal mainitud **img** tagi **emulator** atribuut:

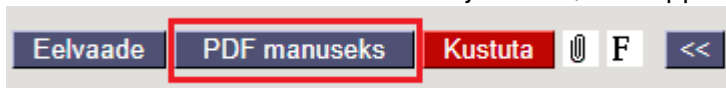
```
<img sign="signhere" emulator="1" width="1" border="0" />
```

PDF manuseks

Kui dokumendile on loodud PDF väljatrükk, mida on vaja digiallkirjastada, saab sellest väljatrükist mugavalt ühe nupu vajutusega luua manuse. Selleks tuleb Seadistused > Väljatrükid alt leida õige dokument, soovitud väljatrüki real teha „Signeeri“ tulbas valik **PDF manuseks** ja salvestada.



Kui eelnev seadistus on tehtud, tekib dokumendile nupp „PDF manuseks“. Olukorras, kus seadistus on tehtud ühe dokumenti mitmele väljatrükile, on nupudel lisaks ka väljatrüki nimetus.



Nupu vajutuse tagajärjel tekib väljatrükist dokumendi  alla manus.

PDF manuseks loomise tugi on hetkel järgmistel dokumentidel:

- Pakkumine
- Tellimus
- Lähetus
- Arve
- Sündmus
- Klient
- Leping
- Liikumine
- Personal

EveryPay Linkpay

[EveryPay Linkpay](#) abil on võimalik arve väljatrükile luua unikaalne makselink.

Alustamine

LinkPay lahenduse kasutamiseks Directos võta ühendust endale sobivaima EveryPay partnerpangaga, milleks on kas LHV, SEB või Swedbank. Sind saab aidata sinu kliendihaldur või täida ise ära [vastava panga taotlusvorm](#). Hinnainfo koos lepingu tingimustega saad otse pangalt. Loe, kuidas täpsemalt kasutada LinkPayd [EveryPay portaalist](#).

Seadistamine

1. EveryPay portaalis tuleb luua **LINKPAY>Lingid** alt link
2. Lingi detailvaates on kuvatud **Lingi token**, mis tuleb kopeerida ning määrata Directo süsteemi seadistustes EveryPay LinkPay lingi token väärtuseks
3. Lingi *Muuda* vaates tuleb **Aktiivne** tulbas valida aktiivseks ainult *Arve number* väli. Kui kasutusel on ka viitenumber ning EveryPay poolt on aktiveeritud *Pangalingi maksed*, tuleks valida ka Viitenumber.
4. Määrata linnud tulbas *URL MUUDETAV* väljadele *Tasumisele kuuluv summa* (*transaction_amount*) ning *Arve nr* (*invoice_number*), Viitenumbri puhul ka Viitenumber (*reference_number*)

AKTIIVNE	KOHUSTUSLIK	KOHANDATUD VÄLJANIMI	VÄLJA VÄÄRTUS	KLIENDI MUUDETAV	URL MUUDETAV
<input type="checkbox"/>		Ettevõtte nimi või muu info Ettevõtte nimi või muu info			
		transaction_amount Summa	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	reference_number Viitenumber	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	invoice_number Dok.number	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



Kui *URL MUUDETAV* tulpa pole näha, pöörduda EveryPay kasutajatoe poole, et see lisada.



Viitenumber saadetakse, kui süsteemiseadistus Transpordi viitenumber on **Arvelt või Kliendilt**



Süsteemiseadistus EveryPay tasuda välja alus võimaldab määrata, milline summa saadetakse makselingile - arve saldo, kliendi saldo või arve tasuda summa

5. EveryPay portaali **SEADED>Üldised seaded** alt kopeerida **API parool** Directo süsteemiseadistuse EveryPay LinkPay API parool väärtuseks
6. Süsteemi seadistus EveryPay LinkPay URL on vaikesi täidetud testkeskkonna aadressiga <https://igw-demo.every-pay.com> Pärast testimist tuleb see täita live teenuse aadressiga <https://pay.every-pay.eu>
7. Link tekib nii arve kui ka tellimuse XMLi, kui väljatrüki definitsiooni aknasse **Väljad** alt on valitud *EveryPay LinkPay link*



See valik tekib automaatselt peale salvestamist, kui lisada *everypay_link* XMLi tag järgmises punktis toodud näite alusel. Analoogselt arvele edastatakse ka tellimuse puhul viitenumber

8. Lingi võib lisada väljatrükile sobivasse kohta, loomise näide:

```
<a><xsl:attribute name="href"><xsl:value-of
select="/documents/document/everypay_link" disable-output-escaping =
"yes"/></xsl:attribute>MAKSA SIIN</a>
```



Makselingi kaasa minevat tasuda summat saab seadistada süsteemi seadistusega EveryPay LinkPay tasuda välja alus.



Vajadusel saab mitme lingi olemasolul luua eraldi loogika, mis valib mis iganes kriteeriumi põhjal sobiva (eeldefineeritud) lingi, mida konkreetsele väljatrükile luuakse. Seadistamiseks kirjuta palun info@directo.ee

HMAC signatuuri loomine välisele lingile

Võimalus on luua signatuur arvest (või tellimusest) välisele osapoolle.

- Signatuur luuakse dokumendi summa täisosa ja numbri omavahel liitmisel eraldatuna & märgiga ning selle signeerimisega osapoolte vahel jagatud salatunnusega **SHA2 256** algoritmi alusel.
- Salatunnus määratakse Directos süsteemiseadistuse Väljatrüki HMAC signatuuri salatunnus väärtuseks.
- Signatuuri saab lisada väljatrükile sobivasse kohta (mingi välise lingi osana):

```
<a><xsl:attribute
name="href">http://www.minukoht.ee/?hmac=<xsl:value-of
select="/documents/document/hmac_signature" disable-output-escaping =
"yes"/></xsl:attribute>Mingi link</a>
```

Näide

Arve nr: 201400285 Summa : 136.30

```
$secret = "85b97cd7a1"; # Directo süsteemiseadistustes salatunnus
$data = "sum=136&num=201400285" #hashitav string NB! arve summas ainult
täisosa
$hmac = hash_hmac("sha256", $data, $secret); #
d62f1b4761ced20b37c189aab95a55fb60e8b4f8a98c4e74194a47622b7a07a
```

Nipid

0 (mitte NaN), kui pole andmeid

- Päisesse numbri formaat paika

```
<xsl:decimal-format name="N" NaN="0" decimal-separator='.' grouping-
separator=' ' />
```

- Kasutamine

```
<xsl:value-of select="format-number(mingi_vali, '0.00', 'N')"/>
```

Mingi bloki X korda joonistamine

- Luua eraldi template, kus on väljund, mida joonistatakse

```
<xsl:template name="kast">
  <xsl:param name="count"/>

  <xsl:choose>
    <xsl:when test="$count &lt;= 0"/>

    <xsl:otherwise>
      <!-- Sisu mida tahame X korda kuvada-->
      <table border="1">
        <tr>
          <td>Mingi sisu</td>
        </tr>
      </table>

      <xsl:call-template name="kast">
        <xsl:with-param name="count" select="$count - 1"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

- Kutsume joonistaja välja nii palju kui vaja

```
<xsl:call-template name="kast">
  <xsl:with-param name="count" select="2"/>
</xsl:call-template>
```

Node-set XML muutuja, sealt unikaalsete kirjete kuvamine

- stylesheet päises peab olema atribuut xmlns:msxsl="urn:schemas-microsoft-com:xslt"
- XML struktuuriga muutuja **andmed** loomine mingi olemasoleva XMLi põhjal

```
<xsl:variable name="andmed" xmlns="">
  <xsl:for-each select="/document/rows/row">
    <rida>
      <artikkel><xsl:value-of select="artikkel"/></artikkel>
      <nimi><xsl:value-of select="nimi"/></nimi>
    </rida>
  </xsl:for-each>
</xsl:variable>
```

!: %%xmlns="" on vajalik Excel-i XML Spreadsheet 2003 formaadis väljundi jaoks (muidu pole võimalik loodud XMList näiteks sum() teha).

- Ülal loodud muutuja andmete kuvamine

- tänu filtrile **[not(artikkel = preceding-sibling::rida/artikkel)]** kuvatakse ainult unikaalsed artiklikoodid

```
<table>
  <xsl:for-each select="msxsl:node-set($andmed)/rida[not(artikkel =
preceding-sibling::rida/artikkel)]">
    <xsl:sort select="artikkel" data-type="text"
order="ascending"/>

    <tr>
      <td>Artikkel: <xsl:value-of select="artikkel"/></td>
      <td>Nimi: <xsl:value-of select="nimi"/></td>
    </tr>
  </xsl:for-each>
</table>
```

dateAdd - kuupäevale mingi päevade arvu lisamine

- XSL- päises peab olema [skripti deklaratsioon](#)
- dateaAdd funktsioon

```
function dateadd(time, paevi)
{
    var d, time, time_spl,aaa, a;
    time_spl=time.split(' ');
    a = time_spl[0].split('.');
    aaa = new Date(a[2], a[1]-1, Number(a[0])+Number(paevi));
    d=String(((aaa.getDate()<9)?'0':'')
)+aaa.getDate()+'.'+((aaa.getMonth()<9)?'0':'') +
String(aaa.getMonth()+1)+'.'+String(aaa.getFullYear());
    return(d);
}
```

- Funktsiooni kutsumine

```
<xsl:for-each select="/documents/document/rows/row">
  <xsl:variable name="paevi"
select="artikkel_andmed/garantii"/>
  <tr>

    <td height="10" valign="top">
      <xsl:value-of select="ribakood"/>
    </td>
    <td> <xsl:value-of
select="artikkel_lisavaljad/lisa[@kood='REALT']"/>&#160;<xsl:value-of
select="string(script:dateadd(string(..../lahetusaeg),
string($paevi)))/>
    </td>
  </tr>
```

```
</tr>
```

Splitter - Mingi tekstilise sisu laiali jagamine separaatori alusel

Parameetrid 1. String, 2. separaator (näiteks tühik ' '), 3. mitmes element tagastada (algab 0-iga, ehk siis esimene tagastatav element on 0)

- XSL- päises peab olema [skripti deklaratsioon](#)
- splitterfunktsioon

```
function splitter( str, splitter, pos ){
    ret = '';
    var arr = str.split(splitter);
    if(arr[pos])
        ret = arr[pos]

    return ret
}
```

- Funktsiooni kutsumine

```
<xsl:for-each select="/documents/document/rows/row">

    <td> <xsl:value-of
select="string(script:splitter(string(nimetus),' ', 1))"/>
    </td>

</tr>
```

dateDiff - kuupäevade vahemiku päevade arvu arvutamine

- XSL- päises peab olema [skripti deklaratsioon](#)
- dateDiff funktsioon (NB! tegemist on VbScript-iga)

```
<msxsl:script language="VbScript" implements-prefix="script">
    <![CDATA[
function ddiff(m,d1,d2)
    ddiff = DateDiff(m,d1,d2)
end function
]]>
</msxsl:script>
```

- Funktsiooni kutsumine

```
<xsl:value-of select="script:ddiff('d',string(aeg1),string(aeg2))"/>
```

- Funktsiooni kutsumine juhul, kui vaja, et arvestaks ka algkuupäeva

```
<xsl:value-of select="script:ddiff('d',string(r_aeg1),string(r_aeg2))
```

```
+ 1"/>
```

Textarea sisu (rea vahedega, näiteks sündmuse kirjeldus) kuvamine ka väljatrükis reavahedega

Kuna reavahetuse ei kajastu HTML-is siis kasutame <pre> tagi.

- tähtis on, et oleks määratud ka konteineri laius, ehk siis see, mille sees soovitud algne sisu asub, peaks omama **width** definitsiooni, selle näite puhul on see <td>
- <pre> puhul on vaikumisi kasutusel ka teine stiil, seega tuleks fondid jms määrata ka <pre> tagile

```
<td valign="top" style="width:700px" >
  <pre style="white-space: pre-wrap; word-wrap: break-word;">
    <xsl:value-of select="/documents/document/sisu"/>
  </pre>
</td>
```

Page break

Et printeris/PDFis tekkiks uus leht:

```
<div style="page-break-before: always;">
```

Summeerimine

Universaalne summeerimise funktsioon, parameetrid:

1. sisend number mida lisatakse
2. grupp (suvaline konteksti kirjeldav string, kui üks summimine siis võib olla näiteks summ)
3. väljund
 - 0 - sisend number
 - 1 - grupi summa
 - 2 - tühjus
 - 3 - grupi summa ilma summat tühjendamata

Väljund on string tüüpi, seega kui on vaja tulemusega arvutada, tuleb see numbriks konvertida enne. 1 ja 2 nullivad ka hetkel grupi.

```
<xsl:stylesheet version="1.0" xmlns:str="http://xslt.org/string"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:msxsl="urn:schemas-
microsoft-com:xslt" extension-element-prefixes="str" xmlns:script =
"http://topxml.com/forum/script" xmlns:x="ignore" exclude-result-
prefixes="script">
  <xsl:output method="html"/>
  <xsl:decimal-format name="ocra" decimal-separator='.' grouping-separator='
```

```
' />

<xsl:for-each select="/documents/document/rows/row[rv=rn]">

  <xsl:choose>
    <xsl:when test="artikkel='VAHESUMMA'">
      <xsl:value-of select="string(script:summer(0,'summ', 1))"/><!-- siin
kuvatakse grupi summa-->
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test="kogus!='' and kogus!=0">
        <xsl:value-of select="format-number( string(script:summer(
number(rv_summa),'summ', 0)) , '### ##0.00', 'ocra')"/> <!-- siin lisatakse
grupile mingi väärtus-->
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>

</xsl:for-each>

<msxsl:script language="JScript" implements-prefix="script">
<![CDATA[

  var sums = [];

  function summer( v, grp, ret ){
    var rt = '';
    if( !sums[grp] ){
      sums[grp] = 0;
    }
    sums[grp]+=Number(v);

    if( ret == 0 )
      rt = ''+v;
    if( ret == 1 || ret == 3 )
      rt = ''+Number(sums[grp]);
    if( ret == 1 || ret == 2 )
      sums[grp] = 0;
    return rt
  }

]]>
</msxsl:script>

</xsl:stylesheet>
```

Palgateatisele puhkusejäagi kuvamine

```
<xsl:variable name="pjaak1" select="substring-before(substring-after(pjaak,
```

```
';'), ';' )"/>  
<xsl:variable name="pjaak2" select="substring-before(substring-  
after(substring-after(pjaak, ';'), ';'), ';')"/>  
<xsl:variable name="pjaak3" select="substring-before(substring-  
after(substring-after(substring-after(pjaak, ';'), ';'), ';'), ';' )"/>  
<xsl:variable name="pjaak4" select="substring-before(substring-  
after(substring-after(substring-after(substring-after(pjaak, ';'), ';'),  
';'), ';'), ';')"/>  
  
<xsl:value-of select="format-number($pjaak2 - $pjaak3 - $pjaak1 - $pjaak4,  
'#####0')"/>
```

From:
<https://wiki.directo.ee/> - **Directo Help**

Permanent link:
https://wiki.directo.ee/pl/yld_print_form?rev=1744623113

Last update: **2025/04/14 12:31**

