EZ AZ OLDAL MÁR NEM ÉRVÉNYES!

A karbantartott doksi az eVIR-en belül érhető el a szerkesztőfelületen.

PDF Markup Language

A PML az eVIR rendszerben előforduló PDF dokumentumok leírására szolgál. Segítségével valósíthatóak meg az egyedi oldalak a programkód módosítása nélkül.

- A PML még részleges fejlesztés alatt van.
- egyes részei még változhatnak a jövőben

Koncepciók

- Egy szöveges file-ből ami hasonlít a html-re készíteni kell egy PDF file-t
- A leírónyelv nem rekurzív, nem olyan mint az XML, vagy a HTML, bár formailag hasonlít rá.

Tag-ek

<PDF> tag

%WIP% Még nem implementált tag, későbbi bővítésnek fenntartva

```
<pdf>dokuentum</pdf>
```

<DOCUMENT> tag

```
<document sajatoldal>
<table space L5% | C90% | L5%>
| Oldal fejléc |
<row> | <page_content> |
<row> | Oldal lábléc |
```

```
<page>
bla bla bla....
<page sajatoldal>
bla bla bla...
```

- Létrehoz egy dokumentum template-t amelynek a neve: sajatoldal.
- Az oldalban egy táblázatnak kell/javasolt lennie
- a (táblázat/oldal)ban kell lennie egy <page_content> tag-nek, ami arra szolgál, hogy a sajatoldal templatet használó oldalak a táblázat ezen cellájába lesznek betördelve
- Elvileg lehet egy PDF-ben több document tag is,
    - (ezek maguktól nem nyomtatódnak ki)
    - a <page> tagnál meg lehet mondani, hogy melyik document template-et használják. alapértelmezett az utolsónak deklarált, vagy az előző oldalon template-je

Tehát egy dokumentum template általános leírása egy teljes oldalt elfoglaló táblázat, amelynek valahol a közepén a <page_content> tag helyére lesz behelyettesítve a tartalom.

A template lehet például egy fejléces, lábléces forma, itt célszerű mindenféle margót és formázást megadni.

A valódi oldalak tartalma ebbe az dokumentum template formázásába lesz betördelve.

Amennyiben nem fér el egy oldalon a tartalom, úgy több oldalra lesz széttördelve, mindegyik oldal megkapva a dokumentum template környezetet.

<PAGE_CONTENT> tag

általában Lásd a DOCUMENT tag-et

a dokumentum template ezen helyére lesz a tartalom behelyettesítve (ez célszerűen egy táblázatcella legyen).

<PAGE> tag

```
<page doctemplate>
...
```

- Egy új oldalt indít.
    - Opcionálisan megadható, hogy melyik document template-et használja. lásd document tag leírása
    - Ha nincs megadva document template <page> akkor az utoljára definiált dokumentum templatet fogja használni.
    - Egy PDF dokumentumban több különböző <page> is megadható, akár más dokumentum template-tel is. Ezeket egymás mögé fogja fűzni.

Font manipulásló tag-ek

Ide tartoznak a szövegek megjelenítéséért formázásáért felelős tag-ek

```
<h> helvetica/ariel
<t> times
<c> courier
<z> ZapfDingbats
<s> Symbol
```

```
Ezekből az utolsó van életben

<10> 10-es betűméret

Ebből az utolsó van életben

Módosító tag-ek <h> <t> <c> -hez, <z><s> implicit **//

<b> bold
</b> nem bold
<i> italic
</i> nem italic
```

- A tag-ek tetszőleges sorrendben követik egymást, nincs nyitó/záró pár.
- tekinthetőek úgy ezek a tag-ek, mint amik módosítják a _kurzor*<stroke>színét</stroke> betűtípusát, betűméretét, stb... * <h> <t> <c> <z> <s> tag-ek. Az adott standard PDF font lesz a _kurzor*fontja.
  - záró tag nincs, bármely tag átváltja az új fontra.
- <9> fontméret tag-ek. Beállítja a szöveg méretét az adott betűméretre.
  - ha több különböző betűméretű szöveg kerül egy sorba, akkor a betűk alapvonala (alja) kerül egyvonalba.
  - záró tag nincs, bármely más betűméret állító tag elállítja.
-     bold és italic be és kikapcsolása
  - tetszőleges sorrendben lehet a _kurzorstílusát vastag vagy dölt betűsre váltani, vagy onnan visszaállítani.
  - a   nem kötelező záró párok hanem önálló tag-ek.
  - Szabályos ez a kód: `normal <b> bold <i> bolditalic </b> italic </i> normal`

Táblázatok és egyéb formázások

%WIP% A táblázatok kezelése további fejlesztések tervét tartalmazza, alapfunkciók elérhetőek.

```
<table stilus L10% | L75% | R5% | R10% >
<head> {x} | {y}{z} | {a}[**]{a} | teteje <bottom> alja </head>
... | ... | ... | ... |
... |  ... | ... | ...
<row> ... | ...
<row>

</table>

{d} <rows [ciklusid]> {f1} | {f2} ... </rows>

{d} <if choiceid f1> feltételes tartalom </if choiceid>
```

Értelmezés:

- <table …: A tag neve
- stilus: Lehetséges értékei: space, box, cbox %WIP%widespace
  - space: nincs vonal rajzolva a táblázathoz (jellemzően ez kell a <document> tag-en belül)
  - box: hagyományos vonalas táblázat. (minden cella körül vonal rajzolódik
  - cbox: olyan táblázat, ahol minden egyes cella egy legömbölyített téglalapnak látszik.
    - <cbox_line> speciális tag, cbox stílusú táblázatban lehet vele cella széles vonalat húzni (szintaxishoz lásd <line>)
  - további stílusok később kerülhetnek implementálásra
- Oszlop meghatározások betűi

    - Igazítás: L Balra, C Középre, R jobbra, %WIP% J justify
    - szélesség kétféleképpen is megadható. % jellel vagy anélkül, más más jelentéssel
        - 10%: teljest táblázat szélességhez képest 10%.
        - 150: pontméretben
- A | jel a leendő táblázat oszlopainak leírását szeparálja. Példánkban 4 oszlopa lesz a táblázatunknak, 10+75+5+10 (százalékos) bontásban

- %WIP% <head>: Fej mezők megadása . Ez jelenleg nincs implementálva
- | táblázat celláinak határolója
    - tábla végén következő táblasorra továblép, emiatt a <row> használata elhagyható
- <row>: Új sor következik (opcionális)
    - függetlenül attól, hogy hány cella volt kitöltve az előző sorban, új sort indít a táblázatban.
- %WIP% „||” (új sor)
- <rows [ciklusid]>: több sor generálása változó segítségével: lásd változók kezelése...
- <if choiceid var>: feltételes PML content a var kulcsú változó értéke szerint.

- <bottom>: Táblázatban is használható tag, ami az adott cellában a további írásokat alulra igazítja.
    - így lehet egy magasabb méretű cella tetejére és aljára is írni tartalmat. miközben a közepe gyakorlatilag üres lehet.
    - A <bottom> tag előtti részek a cella teteje felé lesznek igazítva, a mögötte levő részek a cella aljához lesznek igazítva.
- oldalra is kiterjesztve a működésmód
- %WIP% cellák tartalmának középre igazítása (függőlegesen)
- </table>: ez a tag kilép a táblázatos üzemmódból szövegsoros üzemmódba.
    - Nem szigorúan lezáró pár, korlátozás nélkül lehet a <table> tag-eket egymás alatti más-más sorformátumú táblázatok építésére használni.

A táblázatokban használt igazítások bárhol használhatóaak az igazító tag-ek használatával. Közös jellemzőjük, hogy egyúttal új sor kezdetét is jelentik, aminek az igazítását megadják.

Ez használható a táblázatok fejében megadott igazítások módosítására is, ami így csak az adott cellában érvényes, tehát a következő sorban/oszlopban már nem.

És használható táblázaton kívül folyó szövegek esetén is.

- <left> balra igazít
- <right> jobbra igazít
- középre igazít
- <justify> justify %WIP% egyelőre nincs implementálva, nem használható.

Táblázatok egymásba ágyazásának kisérleti módja a <sub sub_id> illetve </sub sub_id> tag-ek használatával lehetséges, aminek a hatására a sub tag-en belül újabb táblázat adható meg.

Tartalom tag-ek

Ide olyan tag-ek tartoznak, amik valamit megjelenítenek.

```
bla bla bla
<hspace 100>
<vspace 50>
<logo>
<line 2 [d1 d2]>
```

- bla bla bla: szabadon írt szövegek: a szavak tördelődnek igény szerint (sor végénél, cellákban)
- Az újsor, (enter karakter) az sortörést eredményez.
- Üres sorok, whitespacek all collpases, tehát üres helyet nem lehet így csinálni, a whitespace-ek

használhatóak a PML forrásának áttekinthetőbb írásánál.
- <hspace 100>: vízszintes helyfoglalás 100 pont szélességben (és aktuális fontméret magasságban)
- <vspace 50>: függőleges helyfoglalás 50 pont magasságban (0 szélességben)
    - Nemdokumentált, változás joga fenntartva: speciálisan megadott negatív érték esetén csökkenti a sor eddigi számolt magasságát (feljebb kerül). Hatása átmeneti.
- <logo>: Egy céges logo
- <line 2 [d1 d2]>: Egy vízszintes vonal a sor teljes szélességében adott (2) vonalvastagságban. d1 d2 opcionális szám, szaggatott vonal képzéséhez %WIP%

Egyéb

- <verbatim> </verbatim> párok:
    - Ennek a célja az lenne, hogy a párok közötti szöveget ne értelmezze a rendszer bizonyos részeken (tokenizálási folyamat)
    - pl. így lehet megjeleníteni a { } | és whitespace karaktereket megbízhatóan, anélkül, hogy értelmezné a rendszer.
    - esetleges működéssel kapcsolatos anomáliákról kérünk visszajelzéseket.
- funkció nélküli tag
    - Ez akkor használható, ha nem whitespace tokeneket szerenék egymástól megbízhatóan szeparálni, pl: Mezőnév:„{value}” helyett Mezőnév:„{value}”

Változók kezelése

```
{kulcs}{alkulcs}{subalkulcs}
{kulcs2}
{kulcs3}
{listakulcs}[0]{rekordkulcs1}
{listakulcs}[0]{rekordkulcs2}
{listakulcs}[1]{rekordkulcs1}
{listakulcs}[1]{rekordkulcs2}
```

Az adatszerkezet megadott értékei kerülnek behelyettesítésre, ezek az konkrét pdf-et készítő modulonként más más struktúrát alkothatnak

- A nyomtatáskor az adatszerkezetnek megfelelő értékek kerülnek ide
- esetenként hierarchikusan vannak a kulcsok megadva ({kulcs1}{kulcs2})
- Lista jellegű adatok (rekordok) kezelése PML-ben a <rows ciklusid> tag használatával is van lehetőség
- Feltételes jellegű adatok kezelése PML-ben. a <if choiceid car> tag használatvával is van lehetőség

```
{listakulcs} <rows ciklusid> {rekordkulcs1} {rekordkulcs2} ... </rows ciklusid>
```

- A lista feldolgozásához kell egy {listakulcs} adatszerkezettel megadott struktúra, ami egy listát reprezentál, tehát önmagában nem megjeleníthető a tartalma.
- a <rows ciklusid> és </rows ciklusid> tag-ek közötti részt annyiszor iterálja, ahány eleme a listának van.
- Jellemzően ezt egy táblázatban lehet jól használni, de a táblázat használata nem kötelező a működéshez. (lehet pl. listát is így készíteni)
- A ciklusid tetszőleges egyedi szöveg lehet. A teljes PML-ben egyedinek kell lennie. (Ennek a segítségével találja meg pontosan a /rows párját a tag-nek.
    - a ciklusid használata nem kötelező, (régebbi PML-ekkel kompatibilis) ekkor nem használhatóak egymásba ágyazott rows ciklusok.

Példák a szerződés adatszerkezetein:

1: Szerződéstételek:

```
{szerzodes_tetelek} <rows> {sorsz} : {cikkszam} {megnevezes} {calc_brutto} </rows>
```

2: Partner címei:

```
{ext_partner}{cim} <rows> {cimke} : {nev} {cim_irszam} {cim_varos} {cim_cim} </rows>
```

3: partner telefonszámai:

```
<table box R50% L50%> típus | szám {ext_partner}{telefon} <rows> <row> {tipusnev} |
{szam} </rows>
```

```
{hierarchiakulcs} <if choiceid var> tetszőleges PML tartalom</if choiceid>
```

- Feltételes PML megjelenítéséhez kellhet egy `{hierarchiakulcs}` adatszerkezettel megadott struktúra, aminek valamelyik közvetlen kulcsa egy logikainak értelmezhető változója, tehát önmagában nem megjeleníthető a tartalom.
- a `<if choiceid var>` és `</if choiceid>` tag-ek közötti részt akkor jeleníti meg, ha a {var} értéke logikailag igaz.
- Jellemzően ezt esetlegesen opcionális adatokat tartalmazó minták, táblázat teljes részeinek elrejtésére használható. Pl: Megjegyzés cbox.
- A `choiceid` tetszőleges egyedi szöveg lehet. A teljes PML-ben egyedinek kell lennie. (Ennek a segítségével találja meg pontosan a `/if` párját a tag-nek.
  - a choiceid használata kötelező.
- A `var` az eddig kiszámolt változóhierarchiában már létező scalar logikai változó.
  - A `var` kulcs kiértékelése visszaállítja a hierarchiát a korábbi gyökér állapotába, mintha a `{var}` szintaxis került volna felhasználásra.

```
Jelenleg ilyen használatára még nincs idézhető példa.
```

opcionális változók kezelése

Egyes esetekben az adatszerkezet egyes komponensei hiányozhatnak. Ilyenkor azokat a változók, melyek nem biztos, hogy léteznek, speciális szintaxissal kell írni.

Ilyenkor változó kulcsai vagy indexei mögé egy ? (kérdőjel) karaktert kell írni. Az esetlegesen hiányzó komponenstől kezdve az összes további kulcs vagy index esetén használni kell a ? -et.

Ennek hatására nem történik hibajelzés, ha az adatszerkezetből a változó hiányzik, és nem fog semmi megjelenni.

például a szerződés változói esetén

- `{ext_partner}{cim}{posta}[0?]{cim_cim?}` Ha nincs postacím, akkor így lehet írni
- `{ext_partner}{cim}{posta}[1?]{cim_cim?}` Ha nincs második postacím, akkor így lehet írni
- `{ext_partner}{telefon}[0?]{szam?}` ha nincs telefonszam

`[0]{cim_cim}` rész helyett `[0?]{cim_cim?}` szerepel.

Modulok szerint

- SzerzodesPMLVars A szerzodes modul használata szerint

Megjelenítés

- Első körben beolvassa (és tokenizálja) a PML dokumentumot
  - Ebben benne foglaltatik a tokenek (szavak, whitespacek, táblázatelemek, változók behelyettesítve stb...) létrehozása.
  - A rendszer továbbiakban ezzel a token folyammal fog dolgozni.
  - Itt már ismeri a rendszer a tokenek főbb tulajdonságait, (szavak szélessége, hosszúsága)
- Következő körben eltördeli a hosszú sorokat, így a kinyomtatásra váró tartalom (<page>) egy virtuális papírtekercsre kerül rá
  - A táblázat egy sora mindenképpen egy oldalon belülre igyekszik kerülni
- Mellesleg betördeli a dokumentum templateket is az A4-es oldalra. %TODO% csak A4 lehet?
- Utolsó körben a virtuális papírtekercset feldarabolja a dokumentum templatekbe, így egy esetleg több oldalas, fejléces/lábléces papírra készült pdf tartalom lesz az egészből

- Ha nem férne el egy táblázat (vagy szöveg) egy sora egy oldalon, vagy egy hosszú szó egy cellában vagy egy sorban, akkor a program kimenete meghatározatlan.

From:
https://doc.evir.hu/ - **eVIR tudásbázis**

Permanent link:
https://doc.evir.hu/doku.php/spec:pml

Last update: **2023/12/11 13:01**