|  |  |
| --- | --- |
|  | **EÖTVÖS LORÁND TUDOMÁNYEGYETEM**  **INFORMATIKAI KAR**  **INFORMÁCIÓS RENDSZEREK TANSZÉK** |

XML alapú űrlap kitöltő program

|  |  |
| --- | --- |
| **dr. Nikovits Tibor** | **Verebes Gábor** |
| Információs Rendszerek Tanszék | Programozó matematikus |
| Műszaki tanár | Nappali tagozat |

**Budapest, 2013**

Tartalom

[Felhasználói dokumentáció 1](#_Toc343551843)

[Bevezetés 1](#_Toc343551844)

[Tervezési szempontok 2](#_Toc343551845)

[A szoftver telepítése 3](#_Toc343551846)

[A program indítása 4](#_Toc343551847)

[A program használata 6](#_Toc343551848)

[Szerkesztési mód 6](#_Toc343551849)

[A keret rajzolása 9](#_Toc343551850)

[A háttér beállítása 11](#_Toc343551851)

[Űrlapok kezelése 12](#_Toc343551852)

[A vezérlők 12](#_Toc343551853)

[StaticLabel 13](#_Toc343551854)

[XMLLabel 13](#_Toc343551855)

[XMLTextBox 13](#_Toc343551856)

[XMLCombo 13](#_Toc343551857)

[XMLList 14](#_Toc343551858)

[XMLLargeTextBox 14](#_Toc343551859)

[XMLInsertButton 14](#_Toc343551860)

[XMLPager 15](#_Toc343551861)

[XMLSchemaControl 15](#_Toc343551862)

[Az XPath kifejezés kiválasztása 16](#_Toc343551863)

[Az XML Dokumentumok szerkesztése 18](#_Toc343551864)

[Kitöltési mód 18](#_Toc343551865)

[Fejlesztői dokumentáció 20](#_Toc343551866)

[Felhasznált eszközök 20](#_Toc343551867)

[Definiciók 20](#_Toc343551868)

[Feladat specifikációja 20](#_Toc343551869)

[Modulfelbontás 22](#_Toc343551870)

[Core (XMLFormEditorCore) 22](#_Toc343551871)

[Editor (XMLFormEditor) 28](#_Toc343551872)

[MainForm 28](#_Toc343551873)

[EditorForm 29](#_Toc343551874)

[FillForm 29](#_Toc343551875)

[TextView 29](#_Toc343551876)

[ValidationResultDialog 29](#_Toc343551877)

[OptionsForm 29](#_Toc343551878)

[InputTextDialog 29](#_Toc343551879)

[ControlLibrary (XMLDefaultControlLibrary) 30](#_Toc343551880)

[StaticLabel 32](#_Toc343551881)

[XMLLabel 32](#_Toc343551882)

[XMLTextBox 32](#_Toc343551883)

[XMLCombo 32](#_Toc343551884)

[XMLList 33](#_Toc343551885)

[XMLLargeTextBox 33](#_Toc343551886)

[XMLInsertButton 33](#_Toc343551887)

[XMLPager 34](#_Toc343551888)

[XMLSchemaControl 34](#_Toc343551889)

[Tesztelés 35](#_Toc343551890)

[1. Teszteset – Telepítés 35](#_Toc343551891)

[2. Teszteset – A főablak 36](#_Toc343551892)

[3. Teszteset – XML Dokumentumok kezelése 37](#_Toc343551893)

[4. Teszteset – Az eszköztár 39](#_Toc343551894)

[5. Teszteset – Az oldalak 40](#_Toc343551895)

[6. Teszteset – Szerkesztés 41](#_Toc343551896)

[7. Teszteset – Vezérlők 44](#_Toc343551897)

[8. Teszteset – Kitöltési mód 44](#_Toc343551898)

[Összegzés 46](#_Toc343551899)

[Felhasznált irodalom 47](#_Toc343551900)

[Mellékletek 49](#_Toc343551901)

Felhasználói dokumentáció

Bevezetés

A dolgozat célja egy olyan szoftver elkészítése volt, amely XML dokumentumok tartalmának szerkesztésére szolgál könnyen kezelhető felhasználói felületen keresztül.

Számos XML szerkesztő program létezik, de ezek a programok általában az XML szerkezetéhez értő szakemberek számára készültek. Az ilyen programok felülete még a hozzáértők számára sem mindig nyújt kényelmesen kezelhető nézetet az XML dokumentumokhoz.

Sok esetben egyszerű megoldást jelentene, ha a felhasználók közvetlenül az XML fájlok tartalmát tudnák szerkeszteni olyan űrlapokon keresztül, melyek az XML dokumentumok speciális nézetének tekinthetők.

Az űrlapos szerkesztésnek az előnye, hogy az XML dokumentumhoz több nézetet is rendelhetünk, attól függően, hogy melyik részét szeretnénk a dokumentumnak szerkeszteni. Előfordulhat, hogy a logikailag összetartozó adatok nem egymás után vannak az XML dokumentum hierarchiájában, így a hagyományos szerkesztési móddal több oldalt kell lapozni, hogy a megfelelő node-ot megtaláljuk.

Hosszas keresgélés után sem sikerült olyan egyszerűen kezelhető kisalkalmazást találnom, ami lehetővé tenné XML dokumentumok ilyen nézeteken keresztüli szerkesztését.

Az XSLT transzformációk segítségével az XML dokumentumokat akár HTML, XHTML vagy egyéb más formátumokká tudjuk alakítani, ami lehetőséget ad speciális nézetek megjelenítésére. Ugyanakkor ezek statikus nézetek, nem tudjuk az eredeti adatokat megváltoztatni. Az XSLT transzformációkat a programban megjelenített szerkesztési nézetek előállításánál is használhatjuk, mivel ezeket a szerkesztési nézeteket is XML formátumban írjuk le. Azonban az így előállított űrlapok szerkeszthetőek lesznek.

Az űrlapok előállításának egy egyszerűbb módját is biztosítja a szoftver.

Bár az XSLT transzformációk könnyen elsajátítható nyelven készíthetőek, mégsem a legegyszerűbb módja az űrlapok előállításának. Az űrlapokat könnyen elkészíthetjük a program szerkesztési módjában. A szerkesztési módban az eszköztáron található komponenseket el tudjuk helyezni az űrlapokon és a komponensekhez meg tudjuk adni, hogy azok melyik XML node vagy attribútum értékéhez tartoznak.

A programot különböző területeken használhatjuk.

Egyszerű weboldalak vagy internetes áruházak adatbázisát akár XML fájlokban is leírhatjuk. Ekkor egy gyors megoldás lehet ehhez az XML adatbázishoz néhány nézetet rendelni, hogy az oldal üzemeltetője ezen keresztül módosítsa a honlapjának tartalmát.

Szintén gyorsan készíthetünk adatbevitelhez űrlapokat. Ha strukturált módon szeretnénk sok adatot bevinni és erre egy egyszerű megoldásra van szükségünk, definiálunk a tárolandó adatokhoz egy XML szerkezetet, majd hozzárendelünk egy űrlapot.

Továbbá lehetőséget ad a program XML-ben tárolt konfigurációs fájlok szerkesztésére.

Tervezési szempontok

A program tervezésénél fontos szempont volt, hogy a program egyszerű, könnyen áttekinthető, kisméretű de ugyanakkor bővíthető legyen.

Az alkalmazás felhasználásának két megközelítése lehetséges. A szoftver egyik funkciója, hogy már létező, tetszőleges szerkezetű XML fájlokban szereplő értékeket módosíthatunk űrlapokon keresztül. A másik megközelítés ennek az ellenkezője. Adott űrlaphoz szeretnénk XML fájlokat készíteni, és a kitöltött űrlapot ezekben az XML fájlokban szeretnénk strukturáltan tárolni.

Erre a két felhasználási szemléletre a következőket lehet példaként felhozni.

Készülékeknek vagy szoftvereknek a beállításait gyakran tárolják XML fájlokban. Ezeknek a konfigurációs fájloknak a szerkesztésére használhatjuk a programot. Ebben az esetben adott egy XML fájl kötött szerkezettel, a hozzá tartozó séma fájllal. Mi ennek az XML fájlnak az adatait módosíthatjuk és a módosításokat validálhatjuk a program segítségével. Ugyanakkor a programnak akkor is haszna lehet, ha gyors megoldást keresünk strukturált adatok számítógépes rögzítésére. Vegyünk egy tetszőleges papír alapú űrlapot, ezt szeretnénk a számítógép számára később könnyen értelmezhető módon rögzíteni. Ekkor megtervezzük az adatok szerkezetének megfelelő XML fájlt és készítünk hozzá egy űrlapot. Szükség esetén még séma fájlt is használhatunk az adatok ellenőrzésére. Ezt az űrlapot akár olyanok is kitölthetik, akik nincsenek tisztában azzal, hogy mi milyen szerkezetben kívánjuk az adatokat tárolni.

Természetesen mind a két problémára létezik sok szempontból jobb megoldás, mint amit a szoftver kínál. Például készíthetnénk egy speciális programot a konfigurációs fájl szerkesztéséhez vagy egy relációs adatbázisra épülő webes alkalmazást az űrlapok felviteléhez. Ugyanakkor bizonyos esetekben gyorsabb, egyszerűbb vagy költséghatékonyabb megoldást nyújthat az XML editor használata, mert például nincs szükség több száz megabyte-os drága szoftverek telepítésére vagy mélyebb programozói ismeretekre.

Az egyszerűségen kívül a bővíthetőség állt a tervezés középpontjában.

A program nem nyújt minden problémára megoldást, lehetőség van a program bővítésére. Az űrlapokat különböző komponensekből állíthatjuk össze. Ilyen komponens például a beviteli mező (TextBox), legördülő menü (ComboBox), lista vagy egy egyszerű felirat. A komponensek listája viszonylag egyszerűen bővíthető, így ha egy problémát nem tudunk a meglévő komponensekkel megoldani, kiegészíthetjük azokat újabbakkal.

Például ha XHTML fájlok szerkesztéséhez szeretnénk a programot használni, hasznos lehet egy olyan komponens, aminek a segítségével vizuálisan tudjuk a weboldal színeit állítani, és a komponens a kiválasztott színnek megfelelő színkódot beírja a megadott helyre a fájlon belül. Ehhez saját komponenst kell létrehozni.

A tervezésnél a fentieken kívül azt is figyelembe kellett venni, hogy az XML dokumentumok gyakran nagyméretűek, viszont a számítógép erőforrásai végesek. Azt hogy a program alkalmas legyen nagyobb XML dokumentumok kezelésére kisebb kompromisszumok árán kellett elérni.

Arra is figyelni kellett, hogy az XML dokumentum adataira több helyen is lehet hivatkozni. A hivatkozásokat xpath kifejezésekkel kell meghatározni a felhasználónak. Egy adattagra több különböző xpath kifejezéssel is hivatkozhatunk, így lehet, hogy két komponens azonos adattagot ír és olvas. Ekkor, ha az egyikben megváltoztatjuk az adattag értékét, a másikat is frissíteni kell.

A szoftver telepítése

A programot Windows operációs rendszereken futó számítógépekre telepíthetjük. A telepítés előtt szükséges, hogy a *Microsoft .NET Framework 2.0* keretrendszer a számítógépen megtalálható legyen.

A program telepítéséhez indítsuk el a CD-n található setup.exe programot, ami a telepítésekkor megszokott szokásos felületet nyújtja a felhasználó számára.

A telepítő program felkínálja a lehetőséget, hogy kiválasszuk a könyvtárat, ahová az XML editor kerül. Ha nem változtatjuk meg az alapértelmezett beállítást, akkor a "\Program Files\XMLFormEditor\" könyvtár a telepítés célpontja.

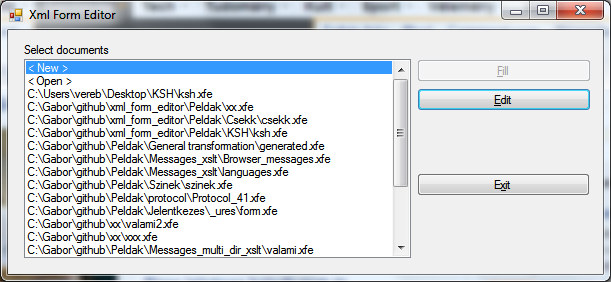
Telepítéskor létrejön az asztalon és a start menüben is egy-egy ikon, amivel a programot indíthatjuk, valamint a \*.xfe kiterjesztésű fájlokhoz hozzárendelődik a kitöltő program, így az ilyen fájlokra ha duplán kattintunk automatikusan elindul a programunk.

A programot a megszokott módón a Windows vezérlőpulton a programok eltávolításánál törölhetjük le.

A program indítása

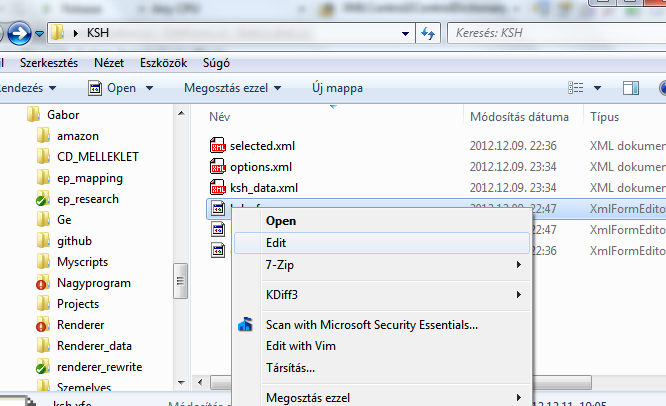
Az XML editort kétféle módban használhatjuk. Az űrlapok létrehozásához a szerkesztési módot, míg az adatok beviteléhez a kitöltési módot kell elindítani.

Ha a programot a start menüből, vagy az asztalról indítjuk, először a dokumentum választó ablak jelenik meg.

Ezen az ablakon azoknak a dokumentumoknak a listáját látjuk, amiket legutoljára nyitottunk meg, de itt tudjuk kiválasztani azt is, ha új dokumentumot szeretnénk létrehozni, vagy a listán nem szereplő dokumentumot szeretnénk megnyitni.

Álljunk rá a lista kiválasztott elemére és nyomjuk meg a *Fill* vagy az *Edit* gombot. A *Fill* gomb hatására az űrlap kitöltés módban, míg az *Edit* gomb megnyomására szerkesztés módban nyílik meg. A *< New >* listaelem kiválasztásánál csak a szerkesztés gombot tudjuk megnyomni. Az alapértelmezett megnyitási mód a kitöltési mód, így ha a lista kiválasztott elemén enter-t nyomunk, a kitöltési mód nyílik meg. A szerkesztési módot billentyűzetről a Ctrl-E gombbal érhetjük el.

A dokumentum választó ablakból az Esc billentyű megnyomásával vagy az Exit gombbal léphetünk ki.

Az űrlapok megnyitásának egy másik kényelmes módja, ha kiválasztjuk a Windows Explorer segítségével a kitölteni vagy szerkeszteni kívánt \*.xfe kiterjesztésű fájlt majd azon jobb gombot nyomunk. A legördülő menüben az *Open* menüponttal megnyithatjuk az űrlapot az alapértelmezett kitöltési módban, míg az *Edit* menüpont segítségével szerkesztési módban nyílik meg az XML Editor.

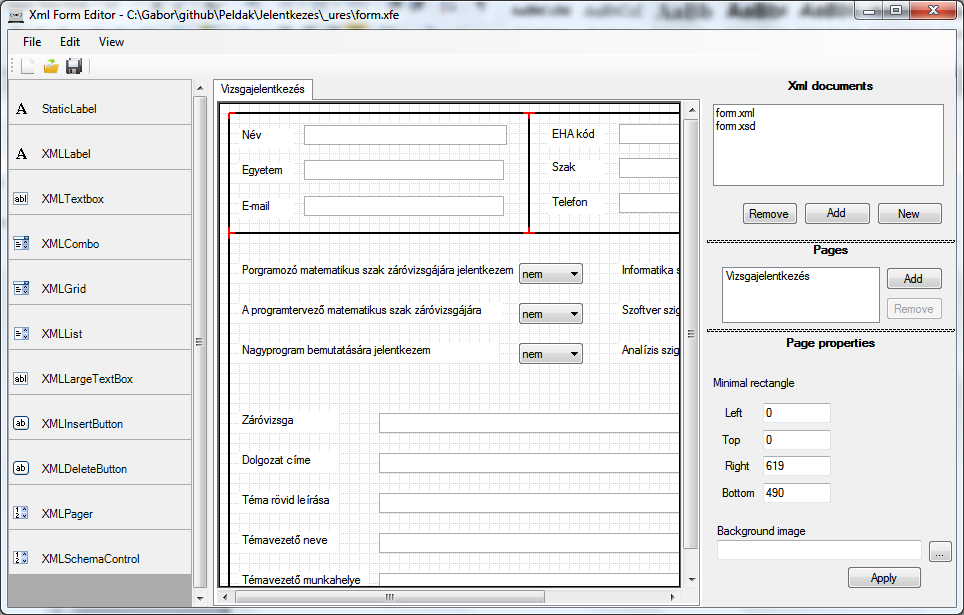
Az XML editort parancssorból is indíthatjuk. Ennek különösen akkor van haszna, ha az XML dokumentumokat vagy az űrlapot leíró \*.xfe fájlt scriptekkel vagy XSLT transzformációkkal hozzuk létre. Ekkor a bemeneti fájlok legenerálását vagy módosítását a script végzi, és az űrlap szolgál az adatok kitöltésére.

Ha parancssorban a futtatható *XmlFormEditor.exe* fájl mögé írjuk a \*.xfe fájl nevét, akkor azt kitöltési módban nyitja meg, míg ha egy *–E* kapcsoló után írjuk a megnyitni kívánt űrlapot, akkor az editor szerkesztési módban indul el. A szokásos jelöléssel:

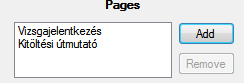
*XmlFormEditor.exe [-E] <fileName.xfe>*

A program használata

Szerkesztési mód

Szerkesztési módban az XML Editor ablaka három részre van felosztva. Az oldal közepén van az űrlap. Az űrlaptól balra van az eszköztár, a felhasználható komponensekkel. Jobb oldalon pedig az űrlaphoz és a kiválasztott komponensekhez tartozó beállítási lehetőségek.

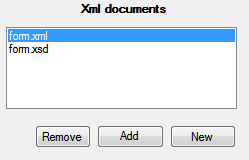
Egy űrlap több oldalból is állhat. Minden oldalnak van egy neve, ami azonosítja az oldalt, így két oldalnak nem lehet azonos az elnevezése. Az oldalakat egyrészt az űrlap felett található fülek segítségével választhatjuk ki, másrészt a jobb oldalon található „Pages” felirat alatt lévő lista segítségével.

Ha új oldalt szeretnénk az űrlapunkhoz adni, vagy egy meglévőt szeretnénk törölni a lista melletti „Add” és „Remove” gombbal tehetjük ezt meg.

Az „Add” gombra kattintva egy dialógus ablakon kell az új oldal nevét megadni. Ha a megadott név már egy meglévő oldal neve, akkor figyelmeztető üzenetet kapunk.

A „Remove” gomb megnyomása után a program megerősítést vár, majd törli a kijelölt oldalt.

Az űrlapokhoz xml dokumentumokat rendelhetünk. Ezeket a dokumentumokat hozzá kell adni a dokumentumlistához, ami a jobb felső sarokban található.

Hozzáadni és törölni az „Add” és „Remove” gombbal tudunk. Egy dokumentumot csak egyszer adhatunk a listához.

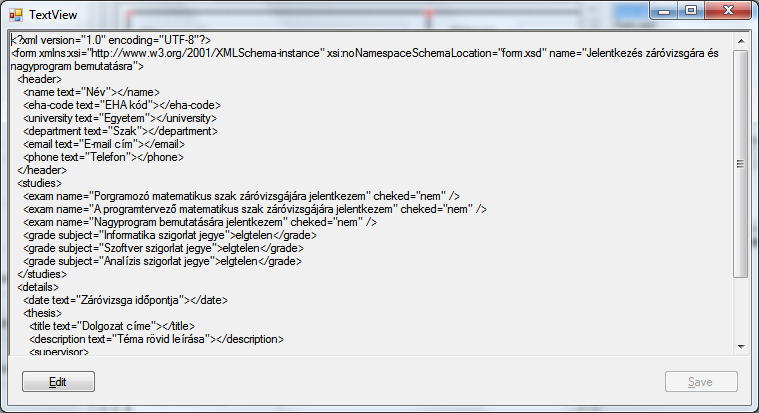
Mivel a dokumentumok az űrlap fájljához relatív útvonalon jelennek meg a listában, először el kell menteni az űrlapot, és csak utána tudunk a listára elemeket felvenni.

Ha új dokumentumot szeretnénk létrehozni, azt is itt tehetjük meg a „New” gomb segítségével. A gomb megnyomásakor ki kell választani egy dialógus ablakon, hogy hova kívánjuk menteni az új xml dokumentumot. Új dokumentumot is csak az űrlap mentése után lehet létrehozni.

A listában szereplő dokumentumokat megtekinthetjük és szerkeszthetjük is.

Ehhez duplán kell kattintani a listában található fájlnéven. Ekkor megnyílik a dokumentum szerkesztésére szolgáló ablak és rajta az xml dokumentum szöveges tartalma csak olvasható formában.

Ha szerkeszteni is szeretnénk a dokumentumot, akkor az „Edit” gombra kell kattintani, aminek a hatására a szerkesztési módba lépünk és az „Edit” felirat „Ok” feliratra változik.

A változásokat az „Ok” gombra való kattintással véglegesíthetjük abban az esetben, ha dokumentum szerkesztésékor az xml dokumentum formátuma érvényes XML formátum marad. Amennyiben a dokumentum érvényessége sérül, például nem zárunk le egy node-ot a fájlban, akkor hibaüzenetet kapunk és szerkesztés módban maradunk.

Sikeres véglegesítés után a módosított dokumentum mentését a „Save” gomb megnyomásával tehetjük meg. Ilyenkor a változások azonnal visszakerülnek az xml dokumentumba a merevlemezen. Ez független attól, hogy mikor mentjük el az űrlapot.

Ha a szerkesztett dokumentumot nem kívánjuk elmenteni, akkor az ablakot zárjuk be a „Save” gomb megnyomása nélkül.

A bal oldalon található eszköztáron találjuk az űrlapon használható vezérlők listáját. Ezeket a vezérlőket egérrel húzhatjuk az űrlapunk kívánt pozíciójára.

Az űrlapon megjelenő vezérlőket mozgathatjuk, átméretezhetjük, törölhetjük, kivághatjuk és másolhatjuk.

A mozgatáshoz rá kell a vezérlőre kattintani, majd a gomb lenyomása mellett arrébb kell húzni a kívánt pozícióra.

Egyszerre több vezérlőt is mozgathatunk. Ehhez ki kell választani a mozgatni kívánt vezérlőket. Kattintsunk a héttérre, majd az egeret lenyomva tartva jelöljük ki azt a téglalapot, amiben található vezérlőket ki szeretnénk jelölni.

A kijelölt elemekhez hozzáadhatunk további elemeket, ha a Ctrl gomb lenyomása mellett rákattintunk a kijelölendő vezérlőre, vagy egy másik téglalapot húzunk.

Ha az összes vezérlőt ki akarjuk jelölni, használjuk a menü „Edit/Select All” menüpontját.

A kijelölést a háttérre kattintva tudjuk megszüntetni, vagy menüből az „Edit/Clear Selection” menüponttal. A kijelölt vezérlőket piros keret jelzi. A piros kereten található kis négyzetek segítségével a vezérlőket átméretezhetjük.

Vannak vezérlők, amelyek méretét széltében és hosszában is állíthatjuk, és vannak olyanok, amelyeket csak horizontálisan módosíthatunk. Az, hogy a vezérlő melyik kategóriába tartozik, a kis négyzetek számából derül ki.

A sarokban található négyzetekkel egyszerre állíthatjuk a szélességet és a magasságot.

Amennyiben több vezérlő méretét szeretnénk egyszerre állítani, a mozgatásnál leírt módon jelöljük ki a vezérlőket és az egyik kis négyzetet húzzuk arrébb. Természetesen a csak horizontálisan módosítható vezérlőket ilyenkor sem tudjuk függőlegesen nyújtani.

A kijelölt vezérlőket másolhatjuk, kivághatjuk, beilleszthetjük és törölhetjük az Edit menü szokásos menüpontjaival (Copy, Cut, Paste, Delete).

A vezérlők elhelyezésének megkönnyítésére szolgálnak a View menü alatt található menüpontok.

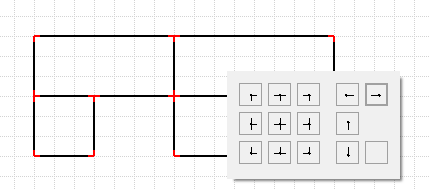
A „View/Arrange left” és „View/Arrange right” menüpontokkal a kijelölt vezérlőket tudjuk egy vonalra igazítani.

A „View/Snap to grid” menüponttal kapcsolható ki és be az a funkció, ami a vezérlők mozgatásánál a vezérlő bal-felső sarkát a háttérben látható rács legközelebbi pontjához igazítja.

Ezt a rácsot ki és be tudjuk kapcsolni a „View/Draw grid” menüponttal, de át is tudjuk állítani a méretét a „View/Options” menü kiválasztásakor felnyíló dialóguson.

Miután elhelyeztük az űrlap különböző elemeit be kell állítani azok paramétereit. Ezt a jobb oldalon található panel alján tehetjük meg akkor, ha csak pontosan egy vezérlő van kiválasztva.

A keret rajzolása

Az űrlaphoz keretet is rajzolhatunk. A keret szerkesztési módban segíthet a vezérlők elhelyezésében, de vizuálisan is strukturálhatjuk az adatainkat a felhasználásával, mivel megjelennek kitöltési módban is.

A keret vízszintes és függőleges vonalakból áll. Két ilyen vonal metszéspontja vagy találkozása a keret egy csomópontja. Az űrlap keretét ilyen csomópontok elhelyezésével rajzolhatjuk meg.

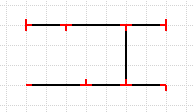
Egy csomópont minimum egy, maximum négy irányt határoz meg, ahonnan másik csomóponttal össze lehet kötni. Ezek az irányok fel, le, jobbra vagy balra indulnak. A csomópontok szerkesztés módban pirossal jelennek meg, kitöltéskor azonban a keret fekete színével.

A csomópontokat az egér bal gombjának hosszan tartó lenyomásával helyezhetjük el. Az egér nyomva tartásakor megjelenő felugró ablakon ki kell választani a csomópont típusát.

Ha az egeret a kiválasztott csomópont típus felett elengedjük, akkor az új csomópont megjelenik az űrlapon.

Amennyiben a csomópont választó ablakot egy meglévő csomópont felett hozzuk elő, akkor módosíthatjuk a csomópont típusát.

A csomópont választó ablakon a kilenc csomópont típus és négy szakasz-végződés közül tudunk választani. Az ablakon található egy üres gomb is. Ha az üres gomb felett engedjük el az egeret, akkor törölhetjük a csomópontot.

A csomópontokat úgy kell elhelyezni, hogy azok egymás irányába mutassanak. Amikor két csomópont egymás felé néz, az űrlap szerkesztő program automatikusan megrajzolja a keret egy szakaszát. Figyeljünk arra, hogy két csomópontnak pontosan egy vonalban kell lennie ahhoz, hogy be lehessen rajzolni a kívánt szakaszt. Amennyiben két csomópont nincs egy vonalban, akár csak egy képponttal eltér a pozíciójuk, a keret szakasza nem jelenik meg. Célszerű a csomópontok elhelyezésekor bekapcsolni a „View/Snap to grid” funkciót, mert ilyenkor a csomópontok automatikusan a rács pontjaira kerülnek.

A csomópontok elhelyezésének egy egyszerűsített módja, ha a „Ctrl+Alt” billentyűket lenyomva kattintunk az űrlapra. Ilyenkor nem kell hosszan nyomva tartani az egér gombját, egyből megjelenik a csomópont választó ablak, ráadásul úgy, hogy a kurzor alatt található az a csomópont típus, ami minden irányba (fel, le, jobbra, balra) mutat. Ha egyből elengedjük az egér gombját, akkor ez, a leggyakrabban használt csomópont kerül az űrlapra.

Egy harmadik módon egyszerre négy csomópontot tudunk elhelyezni, ami egy téglalapot határoz meg. Miközben az „alt” billentyűt lenyomva tartjuk, jelöljünk ki az egérrel egy téglalap alapú területet. Az egér elengedése után négy új, mind a négy irányba mutató csomópont jelenik meg az űrlapon a téglalap négy sarkában. Ezzel a módszerrel gyorsan lehet táblázatszerű kereteket megrajzolni.

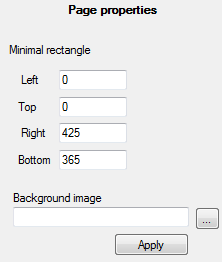
A csomópontok pozícióját utólag is megváltoztathatjuk. Jelöljük ki a mozgatni kívánt csomópontokat. A kijelölés a vezérlők kijelöléséhez hasonlóan történik. A kijelölt csomópontok körül egy piros kör jelenik meg. Egy kijelölt csomóponton nyomjuk le az egér gombját és húzzuk arrébb a kijelölt csomópontokat a kívánt pozícióra.

Több csomópont egyidejű törléséhez jelöljük ki a törlendő csomópontokat és használhatjuk az „Edit/Delete” menüpontot.

A megrajzolt keretet szerkesztés módban arra is használhatjuk, hogy vezérlőket helyezzünk el a kialakított cellákban.

Ha az eszköztárról a keret egy cellájába dobjuk a vezérlőt, akkor a vezérlő mérete a cella mérete lesz és automatikusan kitölti a cellát.

A háttér beállítása

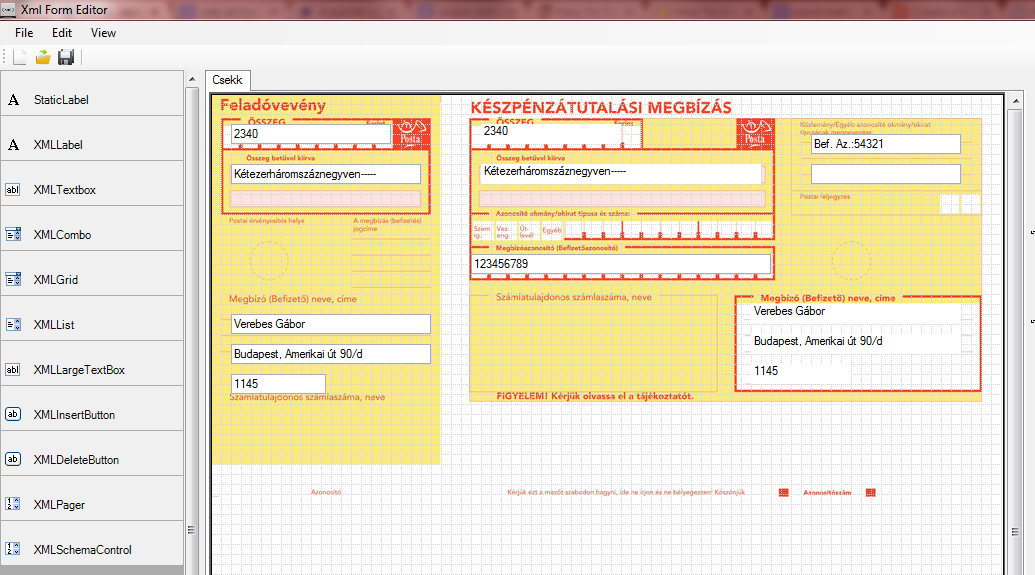
Abban az esetben, ha a háttérre kattintunk és semmi nincsen kijelölve, a vezérlők beállításai helyett az aktuális oldal minimális méretét tudjuk beállítani a jobb oldali panelen. A méret azért minimális, mert ha az űrlap egyik komponensét kihúzzuk a minimális méreten kívülre, akkor az oldal mérete automatikusan megnő akkorára, hogy minden eleme az űrlapnak kiférjen. Az űrlap méretét a left, top, bottom, right értékekkel tudjuk meghatározni, melyek lehetnek negatív számok is.

Az űrlaphoz beállíthatunk egy háttérképet. Ezt a „…” feliratú gombra kattintva tudjuk kiválasztani „Background Image” beviteli mező mellett. A háttérkép betöltéskor automatikusan frissíti a dokumentum minimális méretét.

A változtatások az „Apply” gomb megnyomására lépnek érvénybe.

Ha több egyforma, kitöltött űrlap adatait szeretnénk számítógépre rögzíteni, akkor érdemes egy üres űrlapot képként bedigitalizálni és beállítani háttérként. Helyezzük el a vezérlőket a háttérképen és rendeljünk a vezérlőkhöz XML dokumentumokat.

Az alábbi képen egy postai csekk képére kerültek a beviteli mezők.



Űrlapok kezelése

Az elkészült űrlapot a „File/Save” menüponttal tudjuk elmenteni.

A File menü alatt megtalálható a többi szokásos menüpont is. Itt tudunk új űrlapot létrehozni („New”), meglévőt megnyitni („Open”).

A mentés másként („File/Save As”) menüpont használatakor ügyelnünk kell arra, hogy ha nem ugyanabba a könyvtárba mentjük az űrlapot mint amiben szerkesztettük, akkor a felhasznált xml dokumentumok relatív elérése változhat. Ebben az esetben gondoskodni kell arról, hogy az xml dokumentumok a megfelelő könyvtárba kerüljenek.

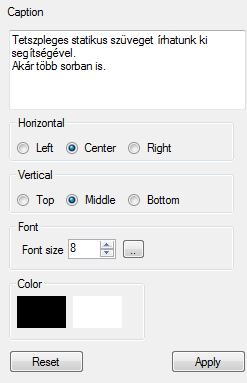
A file menü a fentieken kívül tartalmaz egy „Import” menüpontot.

Ennek a menüpontnak a segítségével az űrlapunkat előállíthatjuk XSLT transzformáció segítségével. Erre például akkor lehet szükség, ha több hasonló űrlapot szeretnénk létrehozni, melyek különböző xml dokumentumokat használnak, vagy az űrlap nagy mérete miatt nem lenne kényelmes annak kézzel történő megszerkesztése.

Importáláskor először meg kell adni egy transzformációt, majd azt az xml-dokumentumot, amin a transzformációt futtatni szeretnénk, és végül azt a fájlt, amibe a beimportált űrlapot menteni kívánjuk.

Az XSLT transzformációk paraméterül kapják a fájlnevet, amin futtatjuk, így azokat fel tudják használni az űrlap generálásakor. A paraméter neve *InputXMLRelativeFileName*.

Az XSLT transzformációkban megengedettek a scriptek és a külső dokumentumok használata.

A vezérlők

Az űrlapok elkészítéséhez a felhasználható vezérlőket a bal oldali eszköztáron találjuk meg. A program tervezésénél fontos szempont volt, hogy a vezérlők listáját könnyen lehessen bővíteni, a vezérlők ezért külön dinamikus könyvtárban (dll) találhatóak. Új dinamikus könyvtár készítésével lehetőség van a vezérlők bővítésére.

A jelenlegi könyvtárban található vezérlők, azok paraméterei és felhasználási módjai a következők:

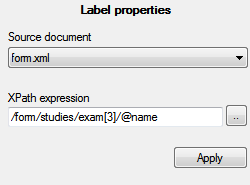
StaticLabel

Tetszőleges, statikus szöveget írhatunk ki segítségével, akár több sorba tördelve. A szöveg tartalmát a vezérlő kijelölése után a jobb oldali panelen megjelenő ablakon írhatjuk be, majd az „Apply” gombra kattintva véglegesíthetjük azt.

A felirat megjelenését megváltoztathatjuk, igazíthatjuk horizontálisan és vertikálisan. Ki lehet választani, hogy milyen betűtípussal jelenjen meg az űrlapon. A felirat színét és a háttérszínét a színeket jelző téglalapra kattintva változtathatjuk meg.

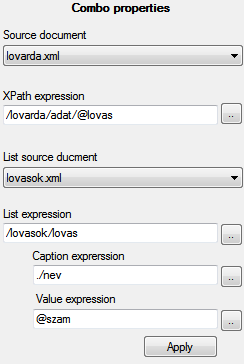
XMLLabel

Szerepe hasonló a StaticLabel-éhez, alkalmazásával feliratokat helyezhetünk el az űrlapon, a kiírandó szöveg azonban egy xml dokumentumból származik, melyet egy xpath kifejezéssel adhatunk meg.

Az xpath kifejezés eredménye lehet egy node az xml dokumentumból, ekkor annak szövegként értelmezett tartalma jelenik meg a feliraton, valamint függvény által visszaadott string érték is. Ha a node vagy a függvény értéke változik a kitöltés közben, a felirat automatikusan frissül.

A jobb oldali panelon ehhez a vezérlőhöz a forrás dokumentumot és az xpath kifejezést kell megadni. Ha érvénytelen kifejezést adunk meg, akkor a felirat helyett figyelmeztető üzenet jelenik meg.

XMLTextBox

Ezzel a vezérlővel tudjuk egy node tartalmát szerkeszteni. Paraméterei megegyeznek az XMLLabel paramétereivel, itt is a forrás dokumentumot és az xpath kifejezést kell megadni, ugyanakkor itt csak olyan xpath kifejezést adhatunk meg, amelynek a visszatérési értéke node típusú.

XMLCombo

Ha legördülő menüt szeretnénk az űrlapunkra helyezni, ezt a vezérlőt kell használnunk.

Paraméterként két xml dokumentumot is meg kell adnunk. Az egyik az a dokumentum, amelyben a kiválasztott érték található, a másik pedig a legördülő menüben szereplő elemeket határozza meg. A két dokumentum azonos is lehet.

Azt a node-ot, ahová a kiválasztott elemhez tartozó értéket írni szeretnénk, úgy adhatjuk meg, mint ahogy azt az XMLTextBox esetében tettük (xml dokumentum + xpath kifejezés).

A legördülő menü elemeihez meg kell adni a menüpontokat tartalmazó xml dokumentumon kívül azt az xpath kifejezést, ami a menüpontokat meghatározó node-ok halmazát adja meg, illetve még két xpath kifejezést, amelyek ezekhez a node-okhoz képest meghatározzák, hogy mi legyen a legördülő menüben megjelenő felirat, és hogy milyen tényleges érték tartozzon ehhez az elemhez. Amikor a kitöltési módban kiválasztunk egy elemet, a kiválasztott elemhez tartozó érték kerül vissza a forrás dokumentumba.

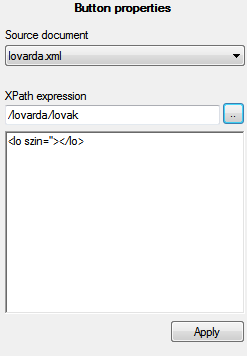
A két utóbbi xpath kifejezés visszatérési értéke lehet node típusú, vagy string típusú, de a menü elemeinek és a hozzájuk tartozó értékeknek egyedieknek kell lennie, azaz két azonos feliratú vagy értékű elem nem szerepelhet a legördülő menüben.

XMLList

Ennek a vezérlőnek a viselkedése és paraméterei megegyeznek az XMLCombo viselkedésével, a különbség csupán annyi, hogy nem legördülő menüt tudunk az űrlapon létrehozni vele, hanem egy lista elemet.

XMLLargeTextBox

Ezzel a vezérlővel tetszőleges hosszú, többsoros beviteli mezőt tehetünk az űrlapra. Ha a szöveg nem fér ki, a gördítő csíkokkal tudunk a szövegben mozogni.

Paraméterei megegyeznek az egyszerű XMLTextBox paramétereivel (xml dokumentum + xpath kifejezés).

XMLInsertButton

Olyan nyomógomb, aminek megnyomására az xml dokumentumba új node-okat szúrhatunk be.

Az xpath kifejezéssel meg kell adni azt a node-ot, aminek végére a gomb megnyomásakor a szintén paraméterként megadandó xml részlet szúródik be.

Az űrlap készítőjének figyelni kell arra, hogy az xml dokumentum helyességét a beszúrt node-ok ne ronthassák el.

XMLPager

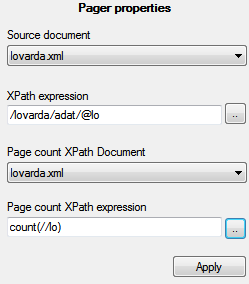
Ennek a vezérlőnek a leírásához nézzük a következő xml dokumentumot:

<dokumentum valasztott="1">  
 <szin>Kék</szin> <szin>Piros</szin> <szin>Zöld</szin>  
</dokumentum>

Tegyük föl, hogy van egy XMLTextBox az űrlapon a következő kifejezéssel:

/dokumentum/szin[number(/dokumentum/@valasztott)]

Ez a kifejezés a *„dokumentum*” node *“valasztott”* attribútumában található sorszámú színt rendeli a beviteli mezőhöz.

Az XMLPager vezérlő ilyen esetekben kínál lehetőséget a lapozásra, azaz a *“valasztott”* attribútum átírására, aminek hatására a beviteli mező automatikusan felveszi az attribútumnak megfelelő új értéket. Természetesen a példa nem csak az XMLTextBox-ra érvényes.

Az XMLPager-nek meg kell adni két xml dokumentumot, melyek lehetnek azonosak is. Az egyik az a dokumentum, ahová az aktuális kiválasztott oldal sorszáma kerül, aminek pozícióját a szintén paraméterként megadott xpath kifejezés azonosítja. A második megadott dokumentumból határozza meg az XMLPager azt, hogy hány oldal van. Ezt egy szám értékkel visszatérő xpath kifejezés adja meg.

XMLSchemaControl

Olyan xml dokumentumok tartalmának szerkesztéséhez melyek szerkezetét séma fájlban írták le, használhatjuk az XMLSchemaControl vezérlőt.

Ennek a vezérlőnek a megjelenése függ a hozzá rendelt node schemában meghatározott típusától. Az űrlapon a vezérlő viselkedhet úgy, mint felirat, beviteli mező vagy legördülő menü. Ha a node típusa a schemában felsorolási típus, a vezérlő olyan legördülő menüként jelenik meg, amiből a node típusa szerinti lehetséges értékeket választhatjuk ki. Ha a felsorolási típus egy elemű, azaz a node csak egy értéket vehet fel (konstans), akkor a vezérlő feliratként viselkedik. Minden egyéb esetben olyan a vezérlő, mintha az XMLTextBox-t használnánk.

A jobb oldali panelen paraméterként kell megadni az xml dokumentumot, az xpath kifejezést, ami a szerkeszteni kívánt node-ot határozza meg, valamint a séma fájlt. Ahhoz, hogy a séma fájlt ki tudjuk választani, hozzá kell adni az űrlapunkhoz, mint xml dokumentumot.

Az XPath kifejezés kiválasztása

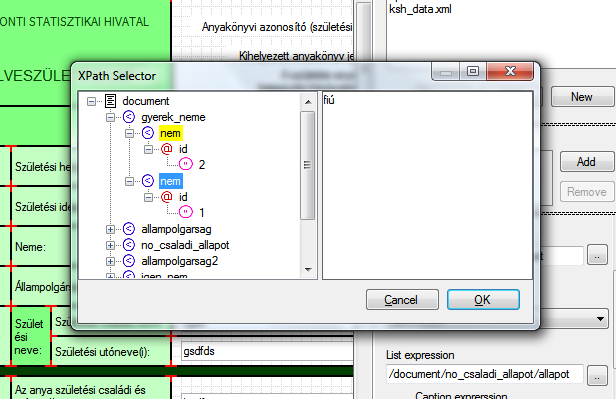
Ahogy a vezérlők tulajdonságainak leírásánál láttuk, a kapcsolatot a dokumentumon elhelyezett vezérlők adatai és az XML dokumentum között XPath kifejezésekkel határozhatjuk meg.

A legegyszerűbb kapcsolatot egy vezérlő és a dokumentum között, amikor egy XPath kifejezéssel kiválasztunk egy node-ot vagy egy attribútumot az xml dokumentumban, hogy onnan vegye és abba tárolja az űrlap a vezérlő értékét.

Van olyan vezérlő is, aminek több XPath kifejezést is meg kell adni. Ilyen vezérlő például az XMLList vagy az XMLCombo.

Ezeknél a vezérlőknél nem csak a kiválasztott érték helyét kell megmondani az XML dokumentumban, hanem azt is, hogy honnan olvassa ki a vezérlő a választható értékeket. Nem biztos, hogy minden esetben a kiválasztott értéket akarjuk tárolni az XML fájlban. Megtehetjük, hogy a lista elemeihez meghatározunk egy értéket ( például azok azonosítóit), hogy az elem kiválasztásakor azt írjuk be az XML dokumentumba. Ilyenkor összesen négy XPath kifejezést kell leírnunk.

Az XPath kifejezések meghatározásához nagy segítséget nyújt az „XPath Selector” dialógus.

Ez az ablak a vezérlők beállításakor jeleníthető meg az XPath kifejezések bevitelére szolgáló mezők mellett található, három pont feliratú gombok megnyomásával.

Az „XPath Selector” megnyitása előtt válasszuk ki, hogy a kifejezés melyik XML dokumentumra vonatkozik.

Amikor a dialógust felnyítjuk az XML dokumentum fa szerkezete jelenik meg. A fa gyökere a dokumentumot reprezentáló node, aminek minden gyermeke az XML dokumentum egy-egy nodja. A további szinteken azok gyermek node-jai vannak.

Amennyiben egy node-nak attribútumai vannak, azok egy „@” ikonnal jelennek meg a fa-szerkezetben. Az attribútumoknak minden esetben pontosan egy gyermeke van, ami az attribútum értékének felel meg. A dialóguson ezeket az attribútum-értékeket egy bekarikázott idézőjel jelöli.

Ha egy node-nak nincsenek gyermek node-jai, akkor a node-hoz tartozó érték láthatóvá válik a fa-struktúra jobb oldalán megjelenő mezőben.

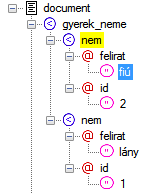
Ez a mező csak akkor látszik, ha az XML dokumentum olyan elemén állunk, amihez rendelhető érték.

Ha a dialóguson kiválasztottuk azt a node-ot, amivel a vezérlőt össze szeretnénk rendelni, zárjuk be az ablakot az enter gomb megnyomásával, vagy az „Ok” gomb megnyomásával.

Az ablak bezárása után a kiválasztott elemhez tartozó XPath kifejezés bekerül ­a vezérlő tulajdonságainak paneljára.

Amikor a dialógust ismét megnyitjuk, akkor az XPath node-nak megfelelő eleme a fának automatikusan lenyílik.

Az „XPath Selector”-on kiválasztott elemekhez a dialógus alap helyzetben olyan kifejezés rendel, ami a dokumentum gyökeréből indul. Amennyiben azonos nevű testvér node-ok vannak a dokumentumban, akkor az XPath kifejezésben a szabványos módon, indexszeléssel különböztetjük meg egymástól ezeket az azonos nevű elemeket.

Ha egy attribútumot vagy annak az értékét választjuk ki, akkor azt attribútumhoz tartozó kifejezést kapjuk, amit a szabvány a „@” jellel különböztet meg a nem attribútum elemektől.

Bizonyos esetekben szükség lehet arra, hogy az XPath kifejezés ne a dokumentum gyökeréhez képest határozza meg az XML dokumentum egy részét.

Ilyenkor megtehetjük, hogy az <F3> gombbal kijelölünk egy elemet az XML fában. A kijelölt elem sárgára vált. Ha ezek után kiválasztunk egy másik elemet és bezárjuk a dialógust, akkor egy olyan XPath kifejezést kapunk, ami a korábban megjelölt elemhez képest határozza meg a kiválasztott node-ot.

Előfordulhat, hogy már a dialógus felnyitásakor van egy megjelölt elem a fán.

Ez akkor történik, amikor például egy XMLCombo vezérlő feliratát vagy annak az értéket akarjuk kiválasztani.

Ilyenkor ugyanis azt feltételezzük, hogy a lista elemek feliratát és értéket a lista elemekhez relatív módon akarjuk megadni. Ha a lista elemét választjuk ki értéknek, akkor egy pont a visszakapott XPath kifejezés, mivel a bejelölt elemhez képest saját magát egy pont-tal azonosítjuk a szabvány szerint.

Az XML Dokumentumok szerkesztése

Amikor az űrlap szerkesztő programot úgy használjuk, hogy egy meglévő űrlaphoz definiálunk egy XML dokumentum szerkezetet, akkor kényelmes, hogy az „XPath Selector” dialóguson nem csak az XPath kifejezést tudjuk kiválasztani, hanem az XML dokumentumot is tudjuk szerkeszteni.

Az XML szerkesztését az alábbi gyorsbillentyűkkel tehetjük meg:

Attribútum beszúrása: Shift + Insert

Elem vagy attribútum nevének szerkesztése: F2

Attribútum értékének szerkesztése: F2

Gyermek node beszúrása: Ctrl + Insert

Testvér node beszúrása: Alt + Insert

Levél elem értékének szerkesztése: Alt + Jobbra nyíl

Vissza az elem szerkesztéséből a fára: Alt + Balra nyíl

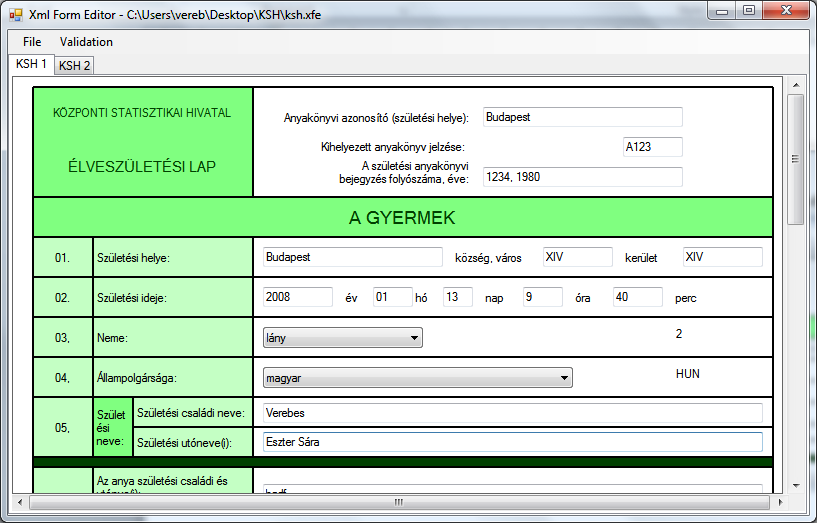
A kiválasztott node és annak összes leszármazottjának törlése: Del

Az egér segítségével a fa részeit áthelyezhetjük egy másik elem alá, de a fenti funkciók is elérhetőek a jobb gombra felnyíló menüből is.

Amikor az „Ok” gombbal vagy egy elem kiválasztásával bezárjuk a dialógust, akkor a módosításokat azonnal belementjük az XML fájlba. Ez később már nem állítható vissza. Ha nem szeretnénk a módosításokat elmenteni, akkor zárjuk be a dialógust a „Cancel” gombbal.

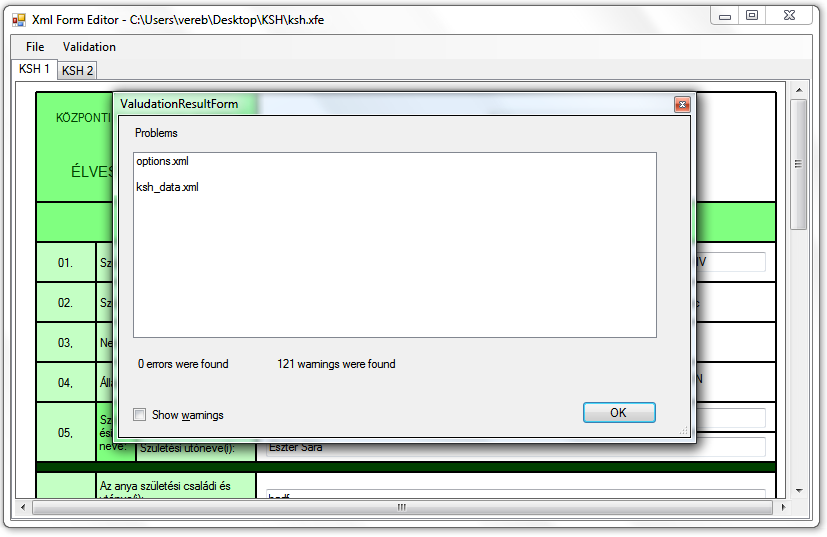
Kitöltési mód

Ha az űrlapot kitöltési módban nyitjuk meg, akkor a megtervezett űrlap jelenik meg, és a vezérlők elérhetővé válnak.

A fenti fülekkel tudunk az oldalak között váltani, hasonlóan, mint a szerkesztés módban.

Módosításainkat a „File/Save” menüponttal, vagy a Ctrl-S megnyomásával tudjuk véglegesíteni. Ekkor kerülnek be ténylegesen a módosítások az űrlap által felhasznált xml dokumentumokba.

Ha meggondoljuk magunkat és szeretnénk a még nem mentett változásokat visszaállítani, az eredeti dokumentumot visszatölthetjük a „File/Reload original” menüponttal.

Amennyiben a dokumentum(ok)hoz tartoznak séma fájlok, akkor az annak való megfelelőséget az F2 („Validation/Validate”) megnyomásával ellenőrizhetjük.

Abban az esetben, ha a kitöltött dokumentum(ok) valamelyike nem megfelelő a séma szerint, az ilyenkor felnyíló ablakon egy hibalistát kapunk. Ha az ellenőrzéskor keletkezett figyelmeztető üzenetekre is kíváncsiak vagyunk, akkor az F3 („Validation/Validate strict”) gombot kell megnyomnunk.

Fejlesztői dokumentáció

Felhasznált eszközök

A program CSharp nyelven készült. A Microsoft Visual Studio 2005 fejlesztőkörnyezetben.

A program a .NET keretrendszer 2.0-ás változatának szolgáltatásait használja, melyek közül kiemelendő a keretrendszer xml, xpath, xml séma és xslt kezelő névterei és osztályai.

Az xml dokumentumok transzformálása, validálása, a node-ok elérése ezeknek az osztályoknak a felhasználásával történik.

A tesztelésnél felhasznált xml dokumentumok, schemák és xslt transzformációk előállításánál bizonyos esetekben külső Xml szerkesztő programot használtam.

Definiciók

**Látható terület (View rectangle):** Az a téglalap, ami az űrlapból éppen látszik.

**Valódi vezérlő (Real control):** Egy olyan *System.Windows.Forms.Control* osztályból származó komponens, ami az űrlapon ténylegesen megjelenik.

**Vezérlő (Control):** Az űrlapokon vezérlők vannak, ezekkel lehet az űrlap adatait megjeleníteni, megváltoztatni. Ha a vezérlő nem látszik, mert nincs az éppen látható területen, a vezérlő nem hoz létre valódi vezérlőt, csak helyettesíti azt. A vezérlők az *XMLControl* osztályból származnak.

Feladat specifikációja

A programnak a következő feltételeknek kell megfelelni:

* Az alkalmazásnak tetszőleges xml dokumentumokhoz kell tudni űrlapokat rendelni anélkül, hogy az eredeti dokumentumokat tartalmilag vagy szerkezetileg módosítani kellene.
* Egy űrlapnak tetszőleges számú és méretű xml dokumentumot kell tudni kezelni.
* Az űralapok elrendezését és az űrlapokon lévő vezérlők kapcsolatát egy xml szerkezetű, xfe kiterjesztésű fájlban kell leírni. Az űrlapokat leíró fájlok relatív útvonallal kell hivatkozzanak a velük összerendelt xml dokumentumokra.
* Az űrlapokon elhelyezhető vezérlők listáját könnyen kell tudni bővíteni, ezért külön dinamikus könyvtárban kell azokat létrehozni megfelelő interface-eket kell hozzájuk definiálni.
* A programnak tartalmaznia kell a leggyakrabban használt vezérlők implementációját (felirat, beviteli mező, lista, legördülő menü, stb…).
* Az űrlapok tetszőleges számú oldalból állhatnak
* Az űrlapokat szerkesztési és kitöltési nézetben kell tudni megnyitni. Ezeket a nézeteket újrafelhasználható komponensekkel kell megvalósítani, hogy más alkalmazásokban is felhasználhatóak legyenek.
* Szerkesztési nézetben egy eszköztárról kell tudni áthúzni a vezérlőket az űrlapra. A vezérlők beállításait és a hozzájuk rendelt adatok elérést egy panelon kell tudni megadni.
* Az adatokat a vezérlőkhöz szabványos xpath kifejezésekkel kell hozzárendelni.
* Szerkesztési nézetben a vezérlőket az űrlapon kijelölni, mozgatni, átméretezni, kivágni, másolni, törölni kell tudni
* A vezérlők között lehet olyan, amiket csak horizontálisan lehet méretezni.
* Több vezérlőt is lehet egyszerre mozgatni, méretezni.
* A vezérlők elhelyezését kényelmi funkciókkal kell megkönnyíteni: háttérben rács (kikapcsolható), illesztés rácspontokra (kikapcsolható), kijelölt vezérlők egy vonalra helyezése.
* Lehetőséget kell biztosítani űrlapok importálására xslt transzformáció futtatásának eredményeként szerkesztési módban.
* Az adatok forrásául szolgáló xml dokumentumokat szerkesztési módban létrehozni és szerkeszteni kell tudni programon belül, illetve a módosított dokumentumok helyességét ellenőrizni.
* Legyen lehetőség a program parancssorból történő indítására szerkesztési és kitöltés módban is.
* A program normál indításánál egy listából tudjuk kiválasztani a korábban megnyitott dokumentumokat.
* Kitöltési nézetben a vezérlők paramétereit és pozícióit nem lehet megváltoztatni.
* Kitöltéskor az xml dokumentumok mentése előtt a dokumentumok jól formázottságát és séma szerinti helyességét kell tudni ellenőrizni.
* A programhoz készüljön installáló alkalmazás.

Modulfelbontás

A program forrása három jól elkülöníthető részből áll. A főprogramból, a főprogram által felhasznált komponenseket, osztályokat és interface-eket tartalmazó modulból és az űrlapokon felhasználható vezérlőkből.

A solution fájlban ezek a modulok külön project-ként szerepelnek, a projectek elnevezései a fenti sorrendnek megfelelően *XMLFormEdior*, *XMLFormEditorCore* és *XMLDefaultControlLibrary*. A solution fájl tartalmaz egy negyedik project-et is, ami a programhoz készült installert tartalmazza, ennek neve *XMLFormEditorSetup*.

A projectek az *XMLFormEditor* nevű közös namespace-t használják.

Core (XMLFormEditorCore)

Ez a project a főprogram által használt űrlap szerkesztő, űrlap kitöltő és az eszköztár komponenseket, valamit az űrlapok adatszerkezetét leíró osztályokat tartalmazza. Ebben a projectben vannak az űrlapon elhelyezhető vezérlőknek megvalósítandó interface-ek is.

Mivel a másik két modul ezekre az osztályokra és interface-ekre épül, ezért tekinthetünk erre a projectre úgy, mint a program magjára.

A project-ben található komponensek úgy lettek megtervezve, hogy felhasználhassuk azokat más programokban is. Például egy kliens-szerver alkalmazásnál, ahol az xml dokumentumok nem fájlból érkeznek, hanem hálózaton keresztül.

A programnak alkalmasnak kell lenni nagyméretű xml dokumentumok kezelésére is, ami azt jelenti, hogy nem hozhatunk létre minden megjeleníteni kívánt xml elemhez az űrlap legenerálásánál egy hagyományos vezérlőt, mert az lassú lenne és az operációs rendszer erőforrásainak végessége is korlátokat szabna.

Ezt a problémát úgy oldja fel a rendszer, hogy a vezérlőket helyettesíti egy struktúrával, ami leírja, hogy a vezérlőnek az űrlapon hol és milyen paraméterekkel kell megjelennie. A helyettesítő struktúrának az **XMLControl** osztályból kell származnia. A dokumentációban vezérlőként általában a helyettesítő struktúrát értjük. A valódi vezérlő akkor jön létre, mikor az bekerül a látható területre.

Az XMLControl típusú objektumokat a **DocumentLayout** osztály kezeli. Ezen az osztályon keresztül lehet a vezérlőket kijelölni, a kijelölt vezérlőket lekérdezni és a kijelölést törölni. Egy adott ponton, vagy téglalap alatt található összes (vagy csak a kijelölt) vezérlők listáját ettől az osztálytól lehet lekérdezni. A vezérlők pozícióját is ezen az osztályon keresztül kell módosítani. Egy DocumentLayout gyakorlatilag az űrlapunk egy oldalának felel meg, ezért az oldal elnevezését és az oldal méreteit is ebben az osztályban kell keresnünk.

Egy űrlap megfelelője a **DocumentLayoutCollection** osztály, ami a DocumentLayout típusú objektumok (az oldalak) gyűjteménye.

Az űrlap betöltését, elmentését és új űrlap létrehozását végezhetjük el a megfelelő tagfüggvény meghívásával. Az űrlaphoz új oldalt adhatunk vagy törölhetünk ezen az osztályon keresztül.

Az űrlapot leíró fájl nevét az osztály LayoutFileName tulajdonságából olvashatjuk ki.

Az űrlapok oldalai, azokon a vezérlők pozíciója és beállításai mentéskor egy xfe kiterjesztésű xml állományba kerülnek. Ennek az xml dokumentumnak a szerkezete a vezérlők egyedi paraméterei nélkül a következő:

<Document>  
 <Source>file1.xml</Source>  
 …  
 <Source>file2.xml</Source>  
 <Form Name="Name 1">  
 <Control Type=”type1” X=”10” Y=”30” Width=”70” Height=”25”>…</Control>  
 …  
 </Form>  
 …  
 <Form Name="Name n">

<Control Type=”type1” X=”10” Y=”60” Width=”70” Height=”25”>…</Control>  
 …  
 </Form>  
</Document>

A gyökér elem mindig a *<Document>*, majd a *<Source>* elemekben fel kell sorolni az űrlapon használt xml dokumentumok neveit. A *<Form>* elemek az űrlap oldalait írják le, a *„Name”* attribútum az oldal elnevezése. A vezérlőket ezen az elemen belül kell felsorolni.

A DocumentLayoutCollection osztály a betöltéskor elkezdi értelmezni a megadott xfe fájlt. Feltölti az XmlSourceDocumentManager singleton osztály egyetlen objektum példányában tárolt forrás dokumentum listát, illetve létrehozza a DocumentLayout típusú objektumokat és meghívja azok betöltő rutinjait. Ha a betöltő rutin sikerrel jár, a DocumentLayoutCollection felfűzi az oldalak listájára az új DocumentLayout objektumot.

A DocumentLayout osztály példányainak meg kell adni egy olyan objektumot, ami megvalósítja az **IControlFactory** interfész egyetlen függvényét:

XMLControl createControl(string controlType);

A vezérlőket az űrlapok betöltéskor ezen az objektumon keresztül hozzák létre.

Végigmennek a *<Form>* alatt található *<Control>* elemeken, és a @Type attribútum értékét átadva visszakapnak egy XMLControl típusú objektumot. Ennek az objektumnak át kell adni az aktuális *<Control>* elemet, és a vezérlő ebből ki tudja olvasni a saját paramétereit.

Látható, hogy az interface használatával kiküszöböltük, hogy ennek a modulnak tudni kelljen arról, hogy milyen típusú vezérlőket hozhatunk létre. Ezen felül lehetővé tettük, hogy a vezérlők létrehozásának logikáját illetve a vezérlők megjelenését külön dinamikus könyvtárban (dll ) tároljuk. Ezzel válik a program bővíthetősége egyszerűvé.

A korábban említett **XmlSourceDocumentManager** osztály felel az űrlap adatainak forrásául szolgáló xml dokumentumok betöltéséért, kezeléséért, séma szerinti ellenőrzéséért és mentéséért. Az osztály egyetlen példányát a statikus Instance() metódussal kaphatjuk meg.

Az osztály két szótár adatszerkezetet tart fent az XMLDocument objektumok és a hozzájuk tartozó fájlnevek kétirányú kereséséhez.

Az XmlSourceDocumentManager ValidateDocuments() metódusa egy hibalistával tér vissza. A lista egyes elemeit a **ValidationResult** osztály írja le. Megtalálható benne, hogy a hiba melyik xml dokumentumban, annak melyik sorának hányadik karakterén van, valamint a hibaüzenet, amit a .net keretrendszer xml séma ellenőrző szolgáltatása ad meg.

A fenti osztályok az űrlapok és az azokhoz tartozó adatok tárolását végzik, míg a most következőek az űrlapok megjelenítését.

A **DocumentVisualizer** osztály az űrlap egy oldalának megjelenítésére szolgál, a DocumentEditor és a DocumentFormVisualizer ősosztálya. Ez az osztály a UserControl-ból származik, így felteheti a programozó az ablakokra a Visual Studio ablak tervező nézetében is. Feladata a DocumentLayout típusú objektumban szereplő XMLControl-okhoz tartozó valódi vezérlők létrehozása és megszűntetése, amikor azok be- vagy kikerülnek abba az ablakkeretbe (*ViewRectangle*), amit a felhasználó ténylegesen lát az űrlapból.

A vezérlők létrehozása a *recreateControls()* metódus hívásakor történik. Ekkor a DocumentVisualizer kiválasztja a látható vezérlőket, majd az XMLControl *CreateEditorControl()* absztrakt metódusának meghívásával megkapja azt a *System.Windows.Forms.Control* típusú valódi vezérlő objektumot, ami kikerül az űrlapra. Ennek a visszakapott vezérlőnek csak a pozícióját és méretét kell a DocumentVisualizer-nek beállítani, minden egyéb paraméterének beállításáért az XMLControl osztály absztrakt metódusait megvalósító leszármazottja felel.

A DocumentVisualizer osztály rajzolja ki az űrlap hátterét is, ezért ebben az osztályban tudjuk az űrlapon megjelenő rácsot ki- és bekapcsolni, valamit annak méretét beállítani.

A DocumentVisualizer osztály leszármazottja a **DocumentFormVisualizer** és a **DocumentEditorVisualizer** osztályok.

Az előbbi az űrlap oldalát kitöltési nézetben jeleníti meg, a vezérlők létrehozását és frissítését definiálja felül az ősének megvalósításához képest.

A szerkesztési nézet megvalósításáról a **DocumentEditorVisualizer** osztály gondoskodik.

A szerkesztési nézet technikai nehézsége, hogy az űrlapoknak úgy kell kinéznie, mintha szerkesztési módban lennénk, de viselkedése teljesen más. Ha a kitöltési nézetben egy vezérlőre kattintunk, akkor az megkapja a fókuszt, míg szerkesztés közben egy piros keret jelenik meg a vezérlő körül, jelezvén, hogy a vezérlő ki lett jelölve. Szintén meg kell oldani a vezérlők mozgatását és méretezését.

A problémára kétféle megoldás létezik. Mindkettő egy újabb UserControl leszármazott osztály definiálásával történik. Ennek az osztálynak egy példánya mindig a dokumentumon lévő látható vezérlők felett kell hogy legyen. Az egyik megoldás, hogy ezt a komponenst átlátszóvá tesszük, és a kijelöléseket rajzoljuk csak ki rajta, míg a másik, hogy a komponens alatt lévő terület bittérképét átmásoljuk a komponens bittérképére és erre rajzoljuk a kijelöléseket. Ez utóbbi megoldás egyben a kirajzolás villogását is megszűnteti.

A program a második megoldást alkalmazza, és a rajzolás felgyorsításának kedvéért a bittérképet a memóriában tartja két rajzolás között, amennyiben a bittérkép mérete nem változott a legutóbbi rajzolás óta.

Az imént leírt komponenst a **DocumentEditorOverlay** osztály valósítja meg, amely egy példányát a DocumentEditorVisualizer tartalmazza. Ez a két osztály kölcsönösen használja egymás publikus tagjait. A szerkesztési nézet feladatait úgy osztják fel egymás között, hogy a DocumentEditorVisualizer kezeli a valódi vezérlőket, míg az egér eseményeket és a kijelölések kirajzolását a DocumentEditorOverlay. Mindkettő osztály megkapja a szerkesztő komponenshez tartozó XMLDocument egy példányát.

Az Overlay komponens lekezeli az egér lenyomását, mozgatását, felengedését és a *CaptureChange* eseményét. Az egér gombjának lenyomásakor a komponens eldönti, hogy ez az esemény a vezérlő kiválasztását, az űrlap kiválasztását vagy a kijelölések szélén lévő kis négyzetek megfogását (amivel a vezérlőt mozgatni tudjuk) indikálja.

Az egér gombjának lenyomása közben történő egér mozgatásának funkciója attól is függ, hogy hol kezdődött a művelet, ezeket az Overlay komponens belső állapota tartalmazza.

A vezérlők átméretezéséhez használt kis négyzetek kirajzolását és a vezérlők átméretezését könnyíti meg a **ResizeTool** osztály, melyet a DocumentEditor és a DocumentEditorOverlay is használ.

A vezérlőkhöz tartozó paraméterek beállítását az **XMLPropertyControlBase** osztályból származó ablakokon lehet beállítani. Az XMLFormEditor project csak az ősosztályt tartalmazza. A vezérlőknek kell a saját tulajdonságaikhoz tartozó ablakot definiálni, ezért a konkrét megvalósításoknak a vezérlők mellett, a vezérlők dinamikus könyvtárában vannak.

Az egyetlen kivétel az **XMLEditorPropertyControl** osztály**,** ez ugyanis nem a vezérlőkhöz tartozik, hanem az űrlap oldalához. Ezen tudjuk megadni az űrlap minimális méretet.

Az XMLPropertyControlBase egy olyan System.Windows.Forms.UserControl leszármazott, ami az **IDataSourceBase** interfacet valósítja meg. Ebből az interface-ből származó újabb interfaceket is definiálhatnak a vezérlők a saját fordítási egységükben. A vezérlők valójában közvetett módon, ezeken az interface-eken keresztül kapják a beállításokat az XMLProperyControBasel-ből származó ablakoktól. Ennek előnye, hogy nem csak ilyen osztályok objektumaitól, hanem más típusú objektumoktól is kaphatják a beállításokat, például tesztelésnél az ablakokat helyettesíthetjük teszt osztályokkal, és automatizálhatjuk a tesztelést.

Ebben a projectben található az eszköztár komponense is, az **XMLEditorToolbox**.

Az eszköztár tartalmazza szerkesztéskor a felhasználható vezérlőket. Mivel a vezérlők listája ebben a projectben még nem ismert, ezt a listát a főprogramban kell kitöltenünk. Az eszköztárra olyan objektumok tehetők, amik megvalósítják az IToolBoxItem interface metódusait. Az eszköztárhoz egy DocumentEditorVisualizer objektumot kell rendelni, amire a vezérlőket húzhatjuk az egér segítségével.

Az eszköztárra a gombokat az IToolBoxItem interface DrawItem metódusával rajzolja ki az osztály. Az egér mozgatásakor a kurzor megváltozik a kiválasztott ikon formájára, amit az interface getToolBoxCursor() metódusával kér le. Amikor elengedjük az egér gombját az űrlap felett, az adott koordinátával meghívja a Duplicate() metódust, ami az eszköztáron lévő elemet másoltatja le az adott koordinátára. Ezzel a megoldással egy vezérlőt megvalósító osztálynak több példányát is érdemes lehet elhelyezni az eszköztáron különböző beállításokkal.

Az eszköztár feltöltését a **ToolBoxInitializer** osztály végzi.



Editor (XMLFormEditor)

Az alkalmazás ablakai találhatóak az XMLFormEditor projectben, amelyből a futtatható állomány fordul.

MainForm

Az alkalmazás nyitó ablaka, ezen találhatóak a korábban megnyitott dokumentumok, aminek a listáját a registry-ben tároljuk a „HKEY\_CURRENT\_USER\Software\XMLFormEditor\recent” ág alatt, szöveges bejegyzésekben, melyek nevei File1,File2,…FileN.

A program indulásakor ellenőrzi, hogy a registry-ben lévő állományok még léteznek e. Amennyiben nem találja valamelyik korábban megnyitott dokumentumot a megadott útvonalon, azt törli a registry-ből, így nem kerül be a dokumentumok listájába.

EditorForm

Az editor ablakán van az eszköztár, egy TabControl, aminek a lapjain az űrlap szerkesztő komponensek vannak és a jobb oldali panel. Meg kell adni a szerkesztő komponenseknek, hogy a vezérlők beállításaihoz tartozó ablakokat ezen a panelen hozza létre.

A TabControl kiválasztott oldalának megváltozásakor pedig át kell állítani az eszköztárhoz tartozó szerkesztő komponenst.

FillForm

Ez az ablak jelenik meg, ha a főablakon a kitöltést nyomjuk meg.

Csak egy TabControl van rajta, aminek az oldalain az űrlap megjelenítő komponensek vannak.

TextView

Ha a szerkesztő ablakon az xml dokumentumok listáján duplán kattintunk egy elemen, akkor ez az ablak nyílik meg. Itt tudjuk a kiválasztott xml dokumentumot megnézni és szerkeszteni.

Az „Ok” gomb megnyomására a dokumentum ellenőrzése történik a ValidateXml() metódusban.

ValidationResultDialog

Ha kitöltés nézetben az „Validation/Validate” vagy „Validation/Validate (strict)” menüt kiválasztjuk, akkor jelenik meg a séma szerinti ellenőrzés eredménye minden forrás xml fájlra.

Az ablakon a „Show warnings” feliratnál kapcsolhatjuk be vagy ki a figyelmeztető üzeneteket.

OptionsForm

Ezen az ablakon tudjuk a szerkesztő komponensek rácsvonalainak távolságát állítani.

InputTextDialog

Ez az ablak az új oldalak nevének bevitelére szolgál.

Az DocumentEditorVisualizer és a DocumentFormVisualizer komponensek nem tartalmaznak scrollbarokat. Ennek a hiánynak a pótlására ezek köré a komponensek köré egy-egy scrollbart implementáló osztályt tartalmaz a project. Valójában az EditorForm és FillForm ablakok ilyen, scrollbarokkal kiegészített komponenseket használnak.

Az alábbi ábra mutatja a projectben található osztályokat és azok lényegesebb metódusait.

ControlLibrary (XMLDefaultControlLibrary)

Az űrlapon felhasználható vezérlőket tartalmazza az XMLDefaultControlLibaray.

Ha szeretnénk újabb vezérlőket biztosítani az űrlapokhoz, ezt a gyűjteményt kell bővíteni, vagy egy ugyanilyen szerkezetű új könyvtárat kell készítenünk.

A vezérlők működését meghatározó XMLControl leszármazottakon kívül a vezérlők beállításaihoz tartozó ablakokat és az azoknak megvalósítandó interface-eket is ebben a projectben kell létrehozni.

A fentieken kívül pontosan egy olyan gyártó osztálynak is lennie kell a projectben, ami a vezérlőket a nevük alapján hozza létre. Ennek az osztálynak az IControlFactory interface-t kell megvalósítania. A projectben a **DefaultControlFactory** osztály valósítja meg ezt az interfacet.

Minden vezérlőnek saját magát kell tudnia egy paraméterül megkapott xml node-ba elmenteni és onnan visszatölteni. A mentés és a visszatöltés a *serializeToXml(…)* és a *deserializeFromXml(…)* virtuális metódusok implementálásával történik.

Ha egy vezérlő tartalma megváltozik, és a változás bekerül az xml dokumentumba, értesíteni kell a többi vezérlőt is, hogy azok frissítsék saját magukat, hiszen megváltozott egy node az xml dokumentumban és az hatással lehet az állapotukra.

A frissítés úgy történik, hogy a szülőnek az *updateVisibleControls()* metódusát hívja a vezérlő az xml tartalom módosítása után. Mivel a vezérlők nem ismerik a saját szüleik típusát, csak annyit várnak el, hogy a szülők megvalósítsák az **IUpdatableWidow** interfacet, ami tartalmazza az imént említett függvény szignatúráját. Ezt az interfacet a XMLFormEditorCore projectben megvalósított *DocumentForm* és *DocumentEditor* osztályok implementálják.

Az *updateVisibleControls()* hívására a szülő meghívja az összes látható XMLControl vezérlőjének, köztük a frissítést kiváltó vezérlőnek, az *UpdateEditorControl(Control EditorControl)* metódusát. Paraméterként megkapják a *System.Windows.Forms.Control* osztályból származó valódi vezérlőket, majd kiolvassák a módosított xml dokumentumból a megjelenítésükhöz szükséges node-okat és frissítik a valódi vezérlők tartalmát. Természetesen ha a vezérlők nem a felhasználói interakció hatására változnak meg, mint például ebben az esetben, akkor ezek a változások már nem indukálhatnak újabb frissítéseket.

Minden vezérlő a fentiek alapján leírt módon viselkedik. A vezérlők jellemzőit tartalmazza a következő lista.

Ezek a jellemzők:

* paraméter beállító ablak típusa
* az ablak által megvalósított interface típusa
* a vezérlőhöz tartozó node szerkezete az űrlapot tároló xml dokumentumban

StaticLabel

* StaticLabelPropertyControl
* IStaticLabelDataSource

<Control Type="StaticLabel" X="54" Y="55" Width="200" Height="20">  
 <Text

Color="WindowText"

Background="128, 255, 128"

Font="Microsoft Sans Serif, 8.25pt"

Alignment="MiddleLeft">

Label  
</Text>  
</Control>

XMLLabel

* SingleDataSourcePropertyControl
* ISingleDataSource

<Control Type="XMLLabel" X="54" Y="102" Width="200" Height="20">  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
</Control>

XMLTextBox

* SingleDataSourcePropertyControl
* ISingleDataSource

<Control Type="XMLTextBox" X="54" Y="147" Width="200" Height="20">  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
</Control>

XMLCombo

* XMLListDataSourcePropertyControl
* IListDataSource

<Control Type="XMLCombo" X="54" Y="191" Width="200" Height="20"> <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <List>  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <CaptionXPath></CaptionXPath>  
 <ValueXPath></ValueXPath>  
 </List>  
</Control>

XMLList

* XMLListDataSourcePropertyControl
* IListDataSource

<Control Type="XMLList" X="54" Y="246" Width="200" Height="100">  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <List>  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <CaptionXPath></CaptionXPath>  
 <ValueXPath></ValueXPath>  
 </List>  
</Control>

XMLLargeTextBox

* SingleDataSourcePropertyControl
* ISingleDataSource

<Control Type="XMLLargeTextBox" X="54" Y="377" Width="200" Height="100">  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
</Control>

XMLInsertButton

* XMLButtonDataSourcePropertyControl
* IButtonDataSource

<Control Type="XMLInsertButton" X="54" Y="513" Width="200" Height="20"> <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <InsertText></InsertText>  
</Control>

XMLPager

* PagerPropertyControl
* IPagerDataSource

<Control Type="XMLPager" X="54" Y="575" Width="200" Height="20">  
 <SourceDocument></SourceDocument>  
 <XPath></XPath>  
 <PageCountDocument></PageCountDocument>  
 <PageCountXPath></PageCountXPath>  
</Control>

XMLSchemaControl

* SchemaPropertyControl
* ISchemaControlDataSource

<Control Type="XMLSchemaControl" X="54" Y="636" Width="200" Height="21">  
 <SourceDocument></SourceDocument>  
 <SchemaFile></SchemaFile>  
 <XPath></XPath>  
</Control>

Tesztelés

A tesztelés célja annak ellenőrzése, hogy az alkalmazás megfelel-e a program specifikációjának.

A teszt több tesztesetből áll, egy teszteset több követelményt is tesztel. A tesztesetekhez tartozhatnak xml, séma és űrlap állományok, amiket a CD-n a TestCases könyvtár alatt találunk. Minden tesztesethez külön könyvtár tartozik. A tesztelés előtt ezt a könyvtárat a merevlemezre kell másolni, és meg kell róla győződni, hogy a fájlok írhatóak.

1. Teszteset – Telepítés

|  |  |
| --- | --- |
| Előfeltétel | Győződjünk meg róla, hogy nincs az XMLFormEditor telepítve a rendszeren |
| Teszt | A telepítő használatával telepítsük fel a programot |
| Elvárt viselkedés | Telepítés után a program megtalálható a startmenüben |
| Teszt | A Windows Explorerben álljunk a database.xfe fájlra és indítsuk el dupla kattintással |
| Elvárt viselkedés | Az alkalmazás kitöltési módban nyílik meg, és betölti a dokumentumot |
| Előfeltétel | Zárjuk be az alkalmazást |
| Teszt | A Windows Explorerben álljunk a database.xfe fájlra a jobb gombbal hozzuk elő a menüt |
| Elvárt viselkedés | A menü második pontja az „Edit” feliratú |
| Teszt | Kattintsunk az „Edit” menüpontra |
| Elvárt viselkedés | A kiválasztott dokumentum szerkesztés módban nyílik meg az alkalmazásban |
| Előfeltétel | Zárjuk be az alkalmazást |
| Teszt | A vezérlőpulton a „Programok telepítése és Törlése” ablakon töröljük az XMLFormEditort |
| Elvárt viselkedés | Az XMLFormEditor alkalmazás törlődik |

2. Teszteset – A főablak

|  |  |
| --- | --- |
| Előfeltétel | Az XMLFormEditor telepítve van a rendszeren |
| Teszt | Álljunk a < New> feliratra |
| Elvárt viselkedés | A Fill gomb nem elérhető, az Edit és Exit gomb elérhető |
| Teszt | Álljunk az < Open > feliratra |
| Elvárt viselkedés | A Fill, az Edit és Exit gomb is elérhető |
| Teszt | A <New> feliraton állva nyomjuk meg az Edit gombot |
| Elvárt viselkedés | Egy üres dokumentum nyílik meg szerkesztés módban |
| Teszt | Mentsük el a dokumentumot és zárjuk be az ablakot |
| Elvárt viselkedés | Ismét megjelenik a főablak és a listában megtalálható az imént elmentett dokumentum |
| Teszt | Nevezzük át a mentett állományt  Válasszuk ki a mentett dokumentum nevét és nyissuk meg szerkesztési vagy kitöltési módban |
| Elvárt viselkedés | Figyelmezető ablak jelenik meg és eltűnik a listából a mentett dokumentum neve |
| Teszt | Válasszuk ki az < Open > elemet a listából és nyomjuk meg a Fill gombot |
| Elvárt viselkedés | Egy „File” dialógus nyílik meg |
| Teszt | Válasszunk ki egy űrlapot |
| Elvárt viselkedés | Az űrlap szerkesztési módban nyílik meg |
| Előfeltétel | Zárjuk be a szerkesztési ablakot. |
| Teszt | Válasszuk ki az < Open > elemet a listából és nyomjuk meg az Edit gombot |
| Elvárt viselkedés | Egy „File” dialógus nyílik meg |
| Teszt | Válasszunk ki egy űrlapot |
| Elvárt viselkedés | Az űrlap kitöltési módban nyílik meg |
| Teszt | Mentsük el az űrlapot másként és zárjuk be a szerkesztési ablakot |
| Elvárt viselkedés | Ismét megjelenik a főablak és a listában megtalálható az imént elmentett dokumentum is |
| Teszt | Lépjük ki az Exit gomb megnyomásával az alkalmazásból  Töröljük a másként elmentett dokumentumot  Indítsuk újra az alkalmazást |
| Elvárt viselkedés | A listából a törölt fájl neve eltűnik, de a többi állomány neve megmarad. |

3. Teszteset – XML Dokumentumok kezelése

|  |  |
| --- | --- |
| Előfeltétel | Nyissunk meg egy új dokumentumot szerkesztés módban |
| Teszt | Nyomjuk meg az “Add” gombot a z xml dokumentumok listáján |
| Elvárt viselkedés | Figyelmeztető üzenet jelenik meg, hogy mentsük el a dokumentumot xml dokumentum hozzáadása előtt |
| Teszt | Nyomjuk meg az “New” gombot az xml dokumentumok listáján |
| Elvárt viselkedés | Figyelmeztető üzenet jelenik meg, hogy mentsük el a dokumentumot xml dokumentum létrehozása előtt |
| Előfeltétel | Mentsük el az űrlapot a merevlemezen a teszt könyvtárába |
| Teszt | Nyomjuk meg az “Add” gombot a z xml dokumentumok listáján |
| Elvárt viselkedés | Egy fájl megnyitására szolgáló ablak jelenik meg. |
| Teszt | Nyomjuk meg a „Mégse” gombot |
| Elvárt viselkedés | Az ablak bezárul és a lista üres marad. |
| Teszt | Adjuk az űrlaphoz az empty.xml dokumentumot |
| Elvárt viselkedés | Hibaüzenetet kapunk és a lista üres marad |
| Teszt | Adjuk az űrlaphoz az invalid.xml dokumentumot |
| Elvárt viselkedés | Hibaüzenetet kapunk és a lista üres marad |
| Teszt | Adjuk az űrlaphoz az valid1.xml dokumentumot |
| Elvárt viselkedés | A listába bekerül a dokumentum neve relatív elérési úttal |
| Teszt | Adjuk az űrlaphoz az valid1.xml valid2.xml valid3.xml dokumentumokat egyszerre |
| Elvárt viselkedés | Egy hibaablakot kapunk, miszerint a valid1.xml dokumentum már létezik.  Ugyanakkor a másik két dokumentum relatív elérése bekerül a listába. |
| Teszt | Mentsük el a sokumentumot |
| Elvárt viselkedés | A dokumentumot (xfe fájl) egy szövegszerkesztővel megnyitva látható, hogy három <Source> node létezik a <Documen> mode alatt.  A három <Source> node a három xml dokumentum relatív útvonalát tartalmazza. |
| Teszt | Kattintsunk duplán a dokumentumok listáján az egyik dokumentumra |
| Elvárt viselkedés | Megnyílik egy ablak, benne az xml dokumentum tartalma csak olvasható formában |
| Teszt | Nyomjuk meg az „Edit” gombot |
| Elvárt viselkedés | A szöveg szerkeszthetővé válik  Az „Edit” gomb felirata „Ok”-ra változik |
| Előfeltétel | Módosítsuk a dokumentumot úgy, hogy az ne legyen érvényes xml formátum |
| Teszt | Nyomjuk meg az „Ok” gombot |
| Elvárt viselkedés | Hibaüzenetet kell kapnunk és szerkesztési módban kell maradnunk |
| Előfeltétel | Módosítsuk a dokumentumot úgy, hogy az érvényes legyen, de ne egyezzen meg az eredeti dokumentummal. |
| Teszt | Nyomjuk meg az „Ok” gombot  Zárjuk be az ablakot a jobb felső ablakon található X gombbal  Kattintsunk ismét duplán a dokumentum nevére a listán |
| Elvárt viselkedés | Nem szabad hibaüzenetet kapni a művelet folyamán  Az újbóli megnyitás után az eredeti dokumentum jelenik meg |
| Teszt | Az „Edit” gomb megnyomása után változtassuk meg a dokumentum tartalmát egy érvényes, de különböző xml tartalomra  Nyomjuk meg az „Ok” gombot  Nyomjuk meg a „Save” gombot  Kattintsunk ismét duplán a dokumentum nevére a listán |
| Elvárt viselkedés | Nem szabad hibaüzenetet kapni a művelet folyamán  Az újbóli megnyitás után a módosított dokumentum jelenik meg |
| Előfeltétel | Zárjuk be a szöveges nézet ablakát  Helyezzünk az űrlapra egy-egy példányt a vezérlőkből a StaticLabel kívételével. |
| Teszt | Nézzük meg egyesével a vezérlők tulajdonságainak beállítására szolgáló ablakon a „Source document”, „List source document”, „Page count XPath document” és „Schema name” legördülő menüket. |
| Elvárt viselkedés | Mindegyik legördülő menü az xml dokumentumok listáját tartalmazza. |
| Előfeltétel | Válasszunk ki a vezérlők közül néhányon legalább egyet minden xml dokumentumból. Ne feledjük a beállításokat érvényesíteni az „Apply” gombbal. |
| Teszt | Válasszunk ki egy xml dokumentumot az xml dokumentumok listáján, és töröljük ki azt a „Remove gombbal” |
| Elvárt viselkedés | A listából a kiválasztott dokumentum eltűnik  A vezérlők ablakain a „Source document”, „List source document”, „Page count XPath document” és „Schema name” legördülő menükből eltűnik a törölt dokumentum.  Azokon a vezérlőkön, ahol a törölt dokumentumot választottuk ki előzőleg, a beállítás eltűnik. |

4. Teszteset – Az eszköztár

|  |  |
| --- | --- |
| Elvárt viselkedés | Az eszköztár 9 vezérlőt tartalmaz: StaticLabel, XMLLabel, XMLTextBox, XMLCombo, XMLList, XMLLargeTextBox, XMLInsertButton, XMLPager, XMLSchemaControl |
| Előfeltétel | Hozzunk létre egy üres dokumentumot a File/New menüponttal |
| Teszt | Helyezzük el az űrlap első oldalán minden vezérlőből egyet  Váltsunk a második oldalra  Helyezzük el az űrlap második oldalán minden vezérlőből egyet |
| Elvárt viselkedés | Miközben húzzuk a vezérlőket, a vezérlők icon-ját és a vezérlő nevét tartalmazó egérkurzort kapunk.  A vezérlőket létre tudjuk hozni, és azok a megfelelő oldalon jönnek létre |
| Előfeltétel | Töröljük az oldalakat úgy, hogy ne maradjon egy sem |
| Teszt | Próbáljunk az űrlap helyére húzni vezérlőket |
| Elvárt viselkedés | Nem történik semmi, nincs hibaüzenet |
| Teszt | Méretezzük át az ablakot úgy, hogy ne férjen ki minden vezérlő az Eszköztárra |
| Elvárt viselkedés | Görgetéskor az eszköztáron lévő elemek a szokásos módon változnak |

5. Teszteset – Az oldalak

|  |  |
| --- | --- |
| Előfeltétel | Hozzunk létre egy üres dokumentumot a File/New menüponttal  Helyezzünk el néhány vezérlőt mindegyik oldalon. |
| Teszt | Váltsunk az oldalak között a füleken és dupla kattintással az oldalak nevein. |
| Elvárt viselkedés | Füleken történő váltásakor az oldalak neve lesz a kiválasztott elem az oldalak neveinek listáján.  A lista elemein való dupla kattintással a kiválasztott fül is változik. |
| Teszt | Töröljük az oldalakat az oldalak listája melletti „Remove” gombbal. |
| Elvárt viselkedés | Törléskor egy megerősítést váró dialógus nyílik meg.  Mindig a kiválasztott elem törlődik a listából. |
| Teszt | Nyomjuk meg az„Add” gombot a lista mellett. |
| Elvárt viselkedés | Egy beviteli mezőt tartalmazó ablak nyílik meg. |
| Teszt | Írjunk be egy nevet az oldalnak, majd nyomjuk meg az „Ok” gombot |
| Elvárt viselkedés | Létrejön egy új oldal a listában és a füleken is. |
| Teszt | Nyomjuk meg az„Add” gombot a lista mellett.  A beviteli mezőbe írjunk be egy létező oldalnevet.  Nyomjuk meg az „Ok” gombot. |
| Elvárt viselkedés | Hibaüzenetet kapunk, miszerint a megadott nevű oldal már létezik.  Az új oldal nem jön létre. |
| Teszt | Helyezzünk néhány vezérlőt minden oldalon.  Mentsük el a dokumentumot.  Indítsuk újra az alkalmazást.  Nyissuk meg a dokumentumot ismét. |
| Elvárt viselkedés | Hibaüzenetek nélkül az elmentett dokumentumnak kell visszatöltődni. |

6. Teszteset – Szerkesztés

|  |  |
| --- | --- |
| Előfeltétel | Hozzunk létre egy üres dokumentumot a File/New menüponttal  Helyezzünk néhány vezérlőt az első oldalra. |
| Teszt | Az “Edit/Select all” menüponttal jelöljük ki az összes vezérlőt az oldalon. |
| Elvárt viselkedés | Az oldalon az összes vezérlő kijelölődik. |
| Teszt | Az „Edit/Clear selection” menüponttal töröljük a kijelöléseket. |
| Elvárt viselkedés | Egy vezérlő sincs kjelölve. |
| Teszt | Jelöljünk ki néhány vezérlőt a kijelölő négyzettel.  Majd tartsuk a ctrl gombot lenyomva és jelöljünk ki újabb vezérlőket. |
| Elvárt viselkedés | A kijelölés helyesen történik, a ctrl lenyomása közben nem törlődhet egy korábban kijelölt vezérlő sem. |
| Teszt | Töröljük a kijelöléseket a háttéren való kattintással. |
| Elvárt viselkedés | A kijelölés eltűnik. |
| Teszt | Jelöljünk ki egyesével vezérlőket szimpla kattintással a ctrl lenyomása közben |
| Elvárt viselkedés | A kijelölés helyesen történik, a ctrl lenyomása közben nem törlődhet egy korábban kijelölt vezérlő sem. |
| Előfeltétel | Győződjünk meg róla, hogy a „View/Snap to grid” menüpont be van kapcsolva. |
| Teszt | Mozgassuk el a kijelölt vezérlőket az egér segítségével. |
| Elvárt viselkedés | A mozgatás közben és végén a vezérlők mindig rácsponton vannak. |
| Előfeltétel | Kapcsoljuk ki a „View/Snap to grid” menüpontot |
| Teszt | Mozgassuk el a kijelölt vezérlőket az egér segítségével. |
| Elvárt viselkedés | A mozgatás közben és végén a vezérlők nem csak a rácsponton vannak. |
| Teszt | A „View/Draw grid” menüponttal kapcsoljuk ki a háttérben látható rácsot |
| Elvárt viselkedés | A háttérben a rács eltűnik |
| Teszt | Kapcsoljuk vissza a „View/Draw grid” menüponttal a rácsot |
| Elvárt viselkedés | A háttérben a rács ismét látható. |
| Teszt | Írjuk át a rács méretét a „View/Options” menüponttal tetszőleges >= 10 értékre |
| Elvárt viselkedés | A rács sűrűsége a beírt értéknek megfelelően változik |
| Előfeltétel | Jelöljünk ki legalább két olyan vezérlőt, aminek a bal oldala nincs egy oszlopon |
| Teszt | Válasszuk ki a „View/Arrange left” menüpontot |
| Elvárt viselkedés | A vezérlők bal oldala egy oszlopba kerül |
| Előfeltétel | Jelöljünk ki legalább két olyan vezérlőt, aminek a jobb oldala nincs egy oszlopon |
| Teszt | Válasszuk ki a „View/Arrange right” menüpontot |
| Elvárt viselkedés | A vezérlők jobb oldala egy oszlopba kerül |
| Teszt | Válasszunk ki egy StaticLabel vezérlőt az űrlapon. |
| Elvárt viselkedés | Megjelenik a kijelölés és rajta 8 kis négyzet |
| Teszt | A kis négyzetek segítségével változtassuk meg a vezérlő méreteit minden irányba |
| Elvárt viselkedés | A négyzetek felett az egér kurzor megváltozik a megfelelő nyílra.  A középső négyzetekkel egyszerre csak egy irányba tudjuk a méretek változtatni. |
| Teszt | Válasszunk ki a vezérlők közül egy StaticLabel és egy XMLLargeTextBox típusút egyszerre. |
| Elvárt viselkedés | Megjelenik a kijelölés mindkét vezérlőn, rajta 8-8 kis négyzettel |
| Teszt | A kis négyzetek segítségével változtassuk meg a vezérlők méreteit minden irányba |
| Elvárt viselkedés | Mindkét kijelölt vezérlő mérete egyformán változik. |
| Teszt | Válasszunk ki a vezérlők közül egy StaticLabel és egy XMLTextBox típusút egyszerre. |
| Elvárt viselkedés | Megjelenik a kijelölés mindkét vezérlőn, a StaticLabel típusún 8, az XMLTextBox típusún 2 kis négyzettel. |
| Teszt | Mindegyik kis négyzetet húzzuk arrébb. |
| Elvárt viselkedés | Mindkét kijelölt vezérlő mérete széltében egyformán változik, függőlegesen az XMLTextBox típusú változatlan marad. |
| Teszt | Jelöljünk ki néhány vezérlőt az egyik oldalon.  A „Edit/Copy” menü választásával másoljuk ki a vezérlőket.  Váltsunk át egy másik oldalra.  Illesszük be a vezérlőket a „Edit/Paste” menüponttal. |
| Elvárt viselkedés | A vezérlők átmásolódnak a második oldalra beállításaikkal együtt. |
| Teszt | Jelöljünk ki néhány vezérlőt az egyik oldalon.  A „Edit/Cut” menü választásával vágjuk ki a vezérlőket.  Váltsunk át egy másik oldalra.  Illesszük be a vezérlőket a „Edit/Paste” menüponttal. |
| Elvárt viselkedés | A vezérlők eltűnnek az előbbi oldalról, majd átmásolódnak a második oldalra beállításaikkal együtt. |
| Teszt | Jelöljünk ki néhány vezérlőt az egyik oldalon.  A „Edit/Delete” menü választásával töröljük a vezérlőket. |
| Elvárt viselkedés | A vezérlők kitörlődnek. |

7. Teszteset – Vezérlők

|  |  |
| --- | --- |
| Előfeltétel | Töltsük be az allatorvosi lo.xfe példa űrlapot, melyen minden vezérlő legalább egyszer szerepel alkalmas beállításokkal. |
| Teszt | Módosítsuk a vezérlők beállításait a jobb oldali panelon. Minden beállítást legalább egyszer változtassunk meg helytelenre, majd vissza érvényesre. |
| Elvárt viselkedés | Ha a vezérlő beállításai érvényesek, akkor a vezérlőhöz rendelt adatok megjelennek rajtuk.  Ha nem érvényesek, akkor kiszürkülnek és/vagy eltűnnek róluk az adatok. |

8. Teszteset – Kitöltési mód

|  |  |
| --- | --- |
| Előfeltétel | Töltsük be kitöltési módban az allatorvosi lo.xfe példa űrlapot, melyen minden vezérlő legalább egyszer szerepel alkalmas beállításokkal. |
| Teszt | Módosítsuk a vezérlők tartalmát az űrlapon. |
| Elvárt viselkedés | Ha az űrlapon olyan xml-nodeot változtatunk, ami más vezérlőkre is hatással van, akkor minden érintett vezérlő tartalma automatikusan megváltozik |
| Teszt | Töltsük vissza az eredeti dokumentumot a „File/Reload original” menüponttal. |
| Elvárt viselkedés | Egy megerősítést kérő ablak jelenik meg, majd visszatöltődik az eredeti dokumentum |
| Teszt | Válasszuk ki a „Validation/Validate” menüpontot közvetlen a visszatöltés után. |
| Elvárt viselkedés | A validálás során talált hibákat megjelenítő ablak nyílik meg, amin látható, hogy sem hiba sem figyelmeztetés nincs. |
| Teszt | Rontsunk el az űrlap második oldalán egy email címet.  Válasszuk ismét a „Validation/Validate” menüpontot |
| Elvárt viselkedés | Ismét megnyílik az előbbi ablak egy hibával , miszerint az email cím nem helyes. |
| Teszt | Módosítsuk ismét a vezérlőket, majd mentsük el a kitöltött űrlapot a „File/Save” menüponttal.  Indítsuk újra a programot ismét ezzel az űrlappal kitöltési nézetben |
| Elvárt viselkedés | A mentett változásoknak meg kell maradni a mentés után |

A program a tesztelés feltételének megfelel, és a tesztesetekben az elvárt módon viselkedik.

Összegzés

A szakdolgozatomban megterveztem, kiviteleztem, leteszteltem és bemutattam egy általános XML alapú űrlap kitöltő programot.

A szoftver tervezése és fejlesztése során megismertem és elsajátítottam a szoftver elkészítéséhez szükséges technikai eszközöket. Felhasználtam különböző XML technológiákat, mint az XPath kifejezések, XSL transzformációk XML séma meghatározások (XSD).

A tervezéskor törekedtem az alkalmazás bővíthetőségére, ezért a megfelelő interfészek meghatározásával három modulra bontottam fel a programot.

A dokumentum szerkesztő– és kitöltő komponensek újrahasznosíthatóak akár egy teljesen más alkalmazásban. A vezérlőket egyetlen dinamikus könyvtár lecserélésével megváltoztathatjuk, kibővíthetjük.

Megoldást találtam olyan problémákra, mint például az űrlapok szerkesztési nézetben történő megjelenítése.

A szerkesztési nézetben ugyanis az űrlapokat úgy kell megjeleníteni, hogy azokon a vezérlők kijelölhetők, mozgathatók legyenek. Ugyanakkor a vezérlők tartalmát közvetlen ne lehessen módosítani.

Számos funkciót építettem a programba, ami megkönnyíti az űrlapok elkészítését és kitöltését.

Az egérrel vezérlőket húzhatunk a ToolBox-ról az űrlapra. A vezérlők elhelyezésekor egy rács segít a pontos pozíció beállításában.

Lehetőség van csomópontok meghatározásával egy keretet rajzolni az űrlapon. Ez a keret azon kívül, hogy érthetőbbé, látványosabbá teszi az űrlapjainkat, felhasználásával a keret celláihoz igazíthatjuk az űrlapon található vezérlőket.

A vezérlőkhöz egy kényelmesen használható XPath kiválasztó ablakot készítettem. Ezen az ablakon az XPath kifejezések kiválasztásán túl, akár az XML dokumentumhoz új elemeket, attribútumokat is adhatunk, vagy az XML dokumentumot át is szerkeszthetjük.

A szoftver használhatóságát több példával is szemléltettem.

Felhasznált irodalom

Cormen, Leiserson, Rivest, Algoritmusok, Műszaki Könyvkiadó, 1999, [883] ISBN-963-16-3029-3

European Computer Manufacturers Association, International Organization for Standardization, ECMA-334, [ISO/IEC 23270:2006](http://standards.iso.org/ittf/PubliclyAvailableStandards/c042926_ISO_IEC_23270_2006(E).zip)[E], C# Language Specification

http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf

2012. december 10.

World Wide Web Consortium (W3C)

Extensible Markup Language (XML) 1.0 (Fifth Edition)

http://www.w3.org/TR/2008/REC-xml-20081126/

2012. december 10.

Microsoft, MSDN Library, .NET Framework 2.0

http://msdn.microsoft.com/en-us/library/aa139623.aspx

2012. december 10.

Microsoft, MSDN Library, Windows Application UI Development (Windows)

http://msdn.microsoft.com/en-us/library/windows/desktop/ff657751(v=vs.85).aspx

2012. december 10.

Object Managment Group, Unified Modeling Language: Infrastructure v2.0

http://schema.omg.org/spec/UML/2.0/

2012. december 10.

Object Managment Group, Unified Modeling Language: Superstructure v2.0

http://schema.omg.org/spec/UML/2.0/

2012. december 10.

World Wide Web Consortium (W3C)

XQuery 1.0 and XPath 2.0 Formal Semantics (Second Edition)

http://www.w3.org/TR/2010/REC-xquery-semantics-20101214/

2012. december 10.

World Wide Web Consortium (W3C)

XSL Transformations (XSLT) Version 2.0

http://www.w3.org/TR/2007/REC-xslt20-20070123/

2012. december 10.

Wagner, B: Hatékony C#, Kiskapu Kiadó, 2005, [283] ISBN- 963-9301-94-9

World Wide Web Consortium (W3C)

W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures

http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/

2012. december 10.

World Wide Web Consortium (W3C)

W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes

http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/

2012. december 10.

Mellékletek



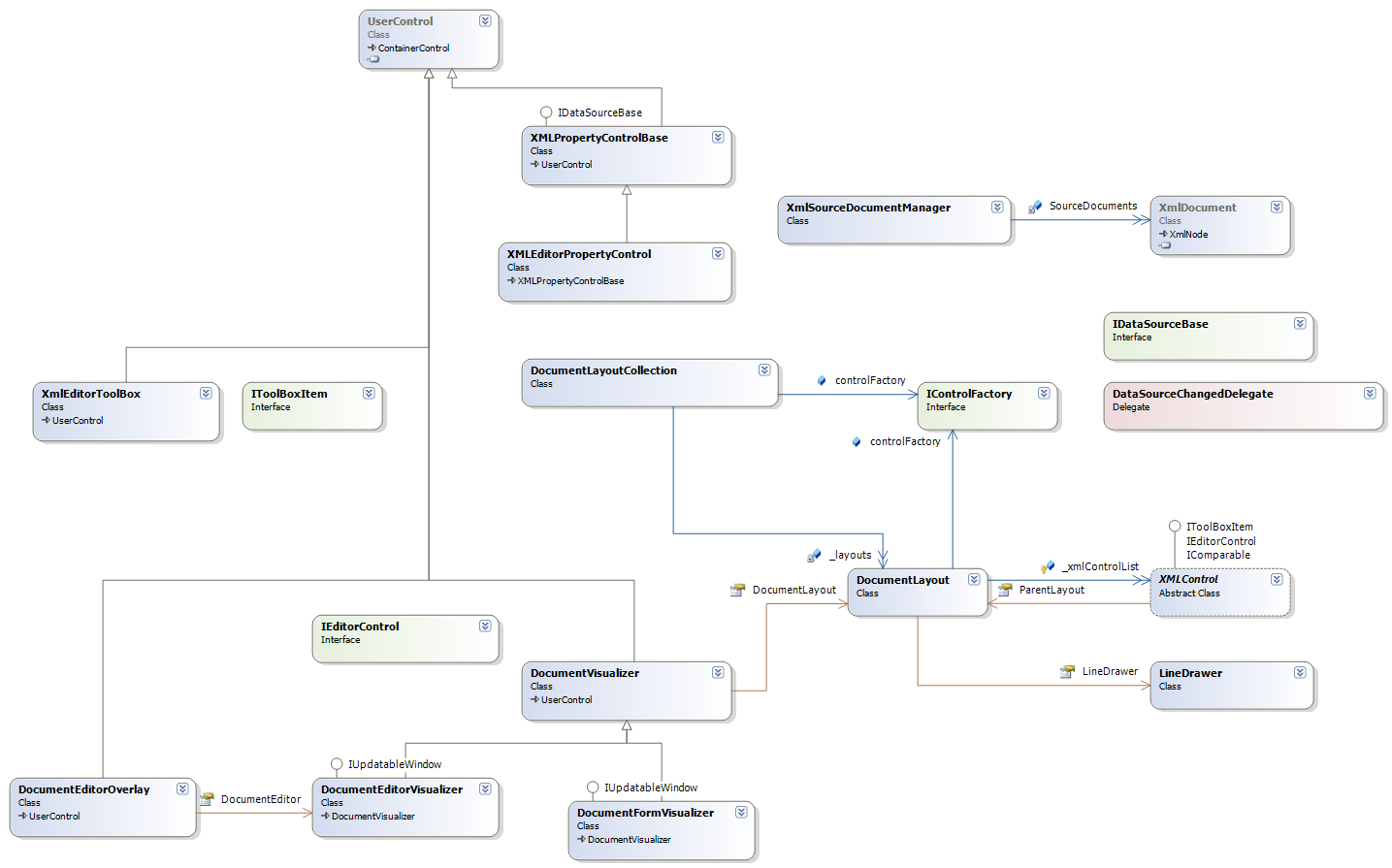
. ábra Vezérlők



2. ábra DataSource Interfacek



3. ábra PropertyPages



. ábra - Az XMLFormEditorCore project