**XSD megtekintő, szótárazó 1.21**

Az Onlineszamla.DLL –be beépített, de annak minden funkciójától függetlenül is használható segédprogramról van szó. Az egyes funkciói egymásra épülnek, és így pl. az InvoiceData.xsd / InvoiceApi.xsd egyszerű megtekintésétől, az egyes mezők definícióinak megjelenítésén át, adatszótár készítéséig sok mindenre használható.

**1., Indítása**

VBNET, C# programkódból :

Utils.ShowViewer()

FoxPro programkódból:

lobridge.InvokeStaticMethod("OnlineSzamla.Utils","ShowViewer","")

Parancssori modulnál:

OlszCommandline.exe XSD

GUI mintaprogramokból



2., Bejelentkező képernyő



Alaphelyzetben XSD állományok feldolgozására készült, de XML állományok is elemezhetőek vele. XML állományok estében csak a szerkezet vizsgálható, XSD esetében a mezőtulajdonságok is megjelennek.

**3., XSD állományok elemzése**

**3.1 InvoiceApi.xsd elemzése**

Az XSD/XML betöltés menüpontra kattintva kitallózzuk a szükséges XSD állományt.



Az állományt a megtekintő beolvassa, lefordítja, így sémaellenőrzést is végez, csak érvényes XSD állományt jelenít meg.



Rögtön látható, hogy az invoiceApi.xsd hány (felső szintű) element elemből áll. Mindegyike megfelel egy-egy hívásnak (request) vagy válaszüzenetnek (response). Az invoiceApi.xsd egyes elemei az invoiceData.xsd-ben vanak definiálva, és onnan importálva, ezért a megtekintő azt is beemelte.

Nyissunk ki a TokenExchangeRequest nódot:



A kötelező adatokat a program vastag betűvel jelzi. Egy adott mezőre tolva az egérkurzort, a tulajdonságok tooltip-ben jelennek meg. A nódokon jobb egérgombbal kattintva egy shortcut menü jelenik meg, és például engedi egy adott node teljes tartalmának klónozását a jobboldali ablakba. Így pl. együtt tekinthetjük át a request és a response definíciót





Jellegéből adódóan az invoiceApi.xsd igen programozóbarát definíció. Kellően denormalizálták, hogy rövid, egyedi funkcióhívásokkal lekódolható legyen. Ugyanez nem mondható el az invoiceData.xsd-ről. Az olyan szinten normalizált, hogy a kódolásához célszerű „útvonal szótárazást” használni. Erre mutatunk egy példát a következőkben.

**3.2 InvoiceData.xsd elemzése**

 Megnyitjuk a már ismertetett módon az invoiceData.xsd állományt. Rögtön szembeszökő a különbség, mert mindössze egyetlen element tagot tartalmaz, igaz abban rögtön két „választás / choice” tipusú elem van, piros szín jelöli. Ez azt jelenti, hogy egy Invoice tagba zárva vagy egy invoiceExchange vagy egy invoiceAnnulment tagot küldhetünk be.



Nyissuk ki mindkettőt:



Jönnek a meglepetések, pl. az invoiceLines nem kötelező, de annak összegzése invoiceSummary kötelező. Kinyitva az invoiceSummary elemet látható egy újabb, ráadásul kötelező választás. Sőt mint látható a summarySimplified többes elem lehet, ezt jelöli a végén, a szögletes zárójel [].



Keressünk további „furcsaságokat”. Klónozzuk a customerinfo nódot:



Azt látjuk, hogy a postalCode nem írja le egyértelműen hova is illesszük be az szállító irányítószámát. Valójában a supplierAddress->simpleAddress->postalCode határozza meg egyértelműen. Azért, hogy se a mostani állapot, se a későbbi változtatás a számla XML és a számlázó program adatmezőinek megfeleltetésében ne okozzon gondot, egy adatszótár generálására alkalmas a megtekintő.



Kattintsunk jobb egérgombbal az invoiceExchange elemen és válasszuk a Szótár generálása menüpontot. A program a kiválaszott nódtól kezdődően legenerálja az utolsó child (adatot hordozó) nódokig terjedő útvonalakat. Ezeket az interfész szótár fül alatt megjeleníti.



Az így elkészült „üres” szótárállomány – milyen meglepetés – XML/XSD formátumban elmenthető a Mentés menüponttal. Utána bármikor visszaolvasható a Beolvasás menüponttal. A generált, mentett XML formátuma és tartalma:

<?xml version="1.0" standalone="yes"?>

<DocumentElement>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceReference.originalInvoiceNumber</QualifyedPath>

<FieldName />

</InvoiceDictionary>

:

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.communityVatNumber</QualifyedPath>

<FieldName />

</InvoiceDictionary>

:

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.customerInfo.customerName</QualifyedPath>

<FieldName />

</InvoiceDictionary>

:

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceLines.line[].quantity</QualifyedPath>

<FieldName />

</InvoiceDictionary>

:

<InvoiceDictionary> <QualifyedPath>invoiceExchange.invoiceSummary.summaryNormal.summaryByVatRate[].vatRate.vatPercentage</QualifyedPath>

<FieldName />

</InvoiceDictionary>

:

<InvoiceDictionary> <QualifyedPath>invoiceExchange.invoiceSummary.summarySimplified[].vatContent</QualifyedPath>

<FieldName />

</InvoiceDictionary>

</DocumentElement>

Visszaolvassuk az állományt, majd elkezdhetjük a saját programunk mező megfeleltetésével összerendelni:



Itt mindenki a saját programnyelvének megfelelő szintaktikát, pl. alias neves relációs szerkezetet, és/vagy leválogatott cursort, stb. használhat. Az eredmény mentés után:

<?xml version="1.0" standalone="yes"?>

<DocumentElement>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierName</QualifyedPath>

<FieldName>szamla.csnev</FieldName>

</InvoiceDictionary>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierAddress.simpleAddress.countryCode</QualifyedPath>

<FieldName>szamla.csorszkod</FieldName>

</InvoiceDictionary>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierAddress.simpleAddress.region</QualifyedPath>

<FieldName xml:space="preserve"> </FieldName>

</InvoiceDictionary>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierAddress.simpleAddress.postalCode</QualifyedPath>

<FieldName>csirsz</FieldName>

</InvoiceDictionary>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierAddress.simpleAddress.city</QualifyedPath>

<FieldName>csvaros</FieldName>

</InvoiceDictionary>

<InvoiceDictionary>

<QualifyedPath>invoiceExchange.invoiceHead.supplierInfo.supplierAddress.simpleAddress.additionalAddressDetail</QualifyedPath>

<FieldName>cscim2</FieldName>

</InvoiceDictionary>

:

</DocumentElement>

**4., XML állomány elemzése**

A fentebb leírt műveletek egy része akkor is elvégezhető, ha XSD állomány nem áll rendelkezésre. Az XSD/XML betöltés menüpontra kattintva kitallózzuk a szükséges XML állományt.



Ebben az esetben a program nem tudja jelezni a kötelezőségeket, és tulajdonságokat, viszont megjeleníti az XML-be foglalt adattartalmat. Működik a nód klónozás. Működik a szótárgenerálás, de nem tudja jelezni a többes nódokat, azokat az elkészült XML-be manuálisan lehet utólag módosítani.

