Wprowadzenie do języków XML, XSD oraz do serializacji dokumentów XML.

Informacje wstępne

Język XML to język znaczników służący do reprezentowania różnego rodzaju danych w ustrukturalizowany sposób.

XSD , czyli XML Schema Definition, służy do definiowania schematu dokumentu XML. Dzięki temu różne instytucje mogą uzgodnić wspólny format wymiany danych.

Oba powyższe języki są technologiami niezależnymi od platformy sprzętowo-programowej.

Dodatkowe informacje:

http://pl.wikipedia.org/wiki/XML

http://en.wikipedia.org/wiki/XML_Schema_(W3C)

http://www.w3.org/XML/

http://pl.wikipedia.org/wiki/Serializacja

1. Pobranie dokumentu XSD.
   XSD jest często stosowanym formatem służącym do opisu danych, zwłaszcza na styku różnych systemów. Przykładem może być tu program SJO BeSTi@, wspomagający samorządy w obowiązkowej sprawozdawczości. Program ten zapisuje dokumenty w postaci XML zgodnej z określonym XSD. Plik XSD o nazwie DokumentPlanistyczny.xsd można pobrać ze strony Ministerstwa Finansów, pod adresem:
   http://www.archbip.mf.gov.pl/bip/5255.html
   Należy wybrać trzeci link od dołu (*dokumentplanistyczny.xsd*), w dziale „Dokumentacja We-Wy systemu SJO BeSTi@". Plik zapisać lokalnie.

2. Generowanie klas z XSD.
   Pakiet Visual Studio jest wyposażony w wiele dodatkowych narzędzi, w tym również w narzędzie do generowania klas na podstawie XSD i odwrotnie. Do wszystkich narzędzi można uzyskać dostęp z poziomu konsoli Visual Studio. Należy uruchomić Start->Microsoft Visual Studio 2010->Visual Studio Tools->Visual Studio Command Prompt. Następnie należy przejść do folderu, gdzie zapisany został dokument XSD I wydać polecenie:

   xsd.exe /c dokumentPlanistyczny.xsd

3. Dodanie klasy do projektu.
   Wygenerowany plik .cs można w kolejnym kroku użyć do łatwego importu i eksportu danych opisanych tym XSD. W tym celu należy uruchomić Visual Studio, utworzyć nowy projekt C# (aplikacja WinForms) i do projektu dodać plik DokumentPlanistyczny.cs. Klikamy PPM (Project and Portfolio Management) na projekcie, Add->Existing Item i wybieramy plik .cs. Od tego momentu możemy posługiwać się wszystkimi klasami wygenerowanymi na podstawie XSD.

4. Import danych poprzez wygenerowaną klasę.
   Pierwszym problemem, jaki trzeba rozwiązać podczas integracji dwóch systemów jest wymiana danych. Tutaj, jak już każdy niewątpliwie zauważył, do opisu formalnego danych służy XSD. Gdyby nie automatyzacja z poprzednich punktów ćwiczenia, to należałoby ręcznie utworzyć klasy odpowiadające konceptom z XSD, następnie odczytać i przeparsować XML, a następnie wypełnić klasę danymi. Na szczęście Bill i jego pracownicy pomyśleli o tym i możemy wykorzystać dostępny w .NET serializator XML, który zaimportuje XML i wypełni klasę danymi za nas. W tym celu:
   a. Dodaj do formy komponent OpenFileDialog (z grupy Dialogs),
   b. We właściwościach tej kontrolki znajdź „Filter" i zapisz tam „Dokument XML|*.xml"
   c. Dodaj do formy komponent PropertyGrid (z grupy All Windows controls),
   d. Dodaj do formy przycisk i kliknij na niego dwukrotnie
   e. W pliku cs głównej formy dodaj dwie dyrektywy using:

```
using System.Xml.Serialization;
using System.IO;
```

   f. W zdarzeniu dodaj kod deserializacji:

```
if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
{
        XmlSerializer xs = new XmlSerializer(typeof(DokumentPlanistyczny));

        using(Stream s = openFileDialog1.OpenFile())
        {
            DokumentPlanistyczny dp = (DokumentPlanistyczny)xs.Deserialize(s);

            propertyGrid1.SelectedObject = dp.Naglowek;
        }
}
```

   Jeżeli wszystko poszło zgodnie z planem, to po otworzeniu dokumentu XML (znajduje się pod linkiem Materiały) kontrolka PropertyGrid wyświetli właściwości pobrane z nagłówka dokumentu.

   g. Na podstawie zaprezentowanych kroków zaprogramuj zapis obiektu do pliku XML. Przedstaw dokonane zmiany odczytując ponownie zapisany (pod inną nazwą) plik XML.