

IGATE Global Solutions Ltd

XML

Lab Guide

Copyright © 2015 IGATE Global Solutions Ltd., Akruti, MIDC Cross Road No. 21, Andheri (E), Mumbai 400 093. All rights reserved. No part of this publication reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of IGATE Global Solutions Ltd.

IGATE Global Solutions Ltd. considers information included in this document to be Confidential and Proprietary.

Table of Contents

Introduction.....	1
Lab Exercise 1: XML Schema Advanced	2
Lab Exercise 2: XSLT,XPath.....	4
Appendix A	7
Appendix B	10

Introduction

This Labbook will guide you through XML. It will help you learn and write your own XML Documents. The Labbook contains solved examples and also the **To Do** assignments. Follow the steps to go through the solved examples and then work out on **To Do** assignments

System Requirements

- Hardware: Networked PCs with minimum 64 MB RAM and 60 MB HDD.
- Software: Window based Operating System having the latest version of Internet Explorer (IE) or Netscape Navigator installed. XML Editor like XML Notepad or XML Spy can be used for parsing you XML document.

Steps to write a XML document:-

Step 1 :- Open Notepad from Start->Program -> Accessories Menu.
(Note :- There are XML Notepad's available to write your XML document)

Step 2 :- Write the code for XML

Step 3 :- Save your file with a suitable filename and .xml extension

Step 4 :- Open Internet Explorer from your Desktop or
Start-> Programs-> Internet Explorer

Step 5 :- In Internet Explorer, select File-> Open Menu. In the dialog box click on browse and select the file you have just saved. Click on Ok.

Step 6 :- Your output is visible in the browser window.

Note:

Create a directory by your name in drive <drive>. In this directory create subdirectory XML_Assgn. For each lab create directory as lab<lab number>

Eg:- d:\XML_Assgn\Lab_Exer1\Prob1.xml

Lab Exercise 1: XML Schema Advanced

Goals	At the end of this lab session you will be able to <ul style="list-style-type: none">➤ Use Advanced XML Schema capabilities
Time	120 min

Consider the following XML file and use Advanced XML Schema concepts to write the XSD for the XML file

```
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <comment>Hurry, my lawn is going wild!</comment>
  <items>
    <item partNum="872-AA">
```

XML LabBook

```
<productName>Lawnmower</productName>  
<quantity>1</quantity>  
<price>148.95</price>
```

```
    <comment>Confirm this is electric</comment>  
  </item>  
  <item partNum="926-AA">  
    <productName>Baby Monitor</productName>  
    <quantity>1</quantity>  
    <price>39.98</price>  
    <shipDate>1999-05-21</shipDate>  
  </item>  
</items>  
</purchaseOrder>
```

Problem 1 :-

Design a XSD file for the above XML file.

Problem 2:-

Redefine the XSD to use a complex type for Address and use it in the XSD file for shipto and billto element.

Problem 3 :-

Consider the same XML file and define namespace for the comment after billto element so that it can be uniquely identified.

Problem 4 :-

Create a separate XSD file for defining address details and then include this in you XSD file.

Lab Exercise 2: XSLT, XPath

Goals	At the end of this lab session you will be able to ➤ Using XPath , XSLT
Time	120 min

Consider the following XML file and apply the XSLT concepts learnt

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog_sort.xsl"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
```

XML LabBook

```

<cd>
  <title>Greatest Hits</title>
  <artist>Dolly Parton</artist>
  <country>USA</country>
  <company>RCA</company>
  <price>9.90</price>
  <year>1982</year>
</cd>
<cd>
  <title>Still got the blues</title>
  <artist>Gary Moore</artist>
  <country>UK</country>
  <company>Virgin records</company>
  <price>10.20</price>
  <year>1990</year>
</cd>
<cd>
  <title>Eros</title>
  <artist>Eros Ramazzotti</artist>
  <country>EU</country>
  <company>BMG</company>
  <price>9.90</price>
  <year>1997</year>
</cd>
<cd>
  <title>One night only</title>
  <artist>Bee Gees</artist>
  <country>UK</country>
  <company>Polydor</company>
  <price>10.90</price>
  <year>1998</year>
</cd>
<cd>
  <title>Sylvias Mother</title>
  <artist>Dr.Hook</artist>
  <country>UK</country>
  <company>CBS Records</company>

```

XML LabBook

```
<price>8.10</price>
<year>1973</year>
</cd></catalog>
```

Problem 1 :-

Display details of those titles which are published after 1990

Problem 2 :-

Display the XML data in a tabular format and highlight the entire record in green for those titles which are published by 'CBS Records' else highlight them in yellow.

Create a XML markup for student data. Use following fields.

Sample of Student data:-

```
<students>
  <student>
    <rollno>1001</rollno>
    <fname>Neena</fname>
    <lname>Kochhar</lname>
    <standard div='A'>IX</standard>
    <firstterm>85</firstterm>
    <secterm>83</secterm>
    <grade>A</grade>
  </student>
```

Considering the above created XML file. Solve the following using XQuery:

Problem 1:

Display student data where they have scored more than 75 in the first term.

Problem 2:

Display all the student detail whose Grade is A sorted on their firstname

Problem 3:

Convert to display the fname and lname of all the students in HTML format and rollno should be added as attribute to fname.

Appendix A

XML Editors :-

1. XML Notepad :- XML Notepad is a simple application that assists in the building of rapid prototypes of XML applications. It allows authors to rapidly build and edit small sets of XML data as a test bed during the development of XML-based applications. With XML Notepad, you can create XML document prototypes quickly, easily, and in an iterative fashion, using familiar metaphors. XML Notepad offers an intuitive and simple user interface that graphically represents the tree structure of XML data. Working with the standard building blocks of XML (Elements, Attributes and Text), authors are able to create reproducible data structures that can be easily filled.

You can download XML Notepad from the Internet. For installation of XML NotePad just execute the Self Extractor Downloaded file. XML NotePad get installed on your PC.

It's an easy to install editor.

To open XML NotePad

From the Programs Menu choose Microsoft XML Notepad → Microsoft XML Notepad

The screen for XML Notepad appears as follows:-

XML LabBook

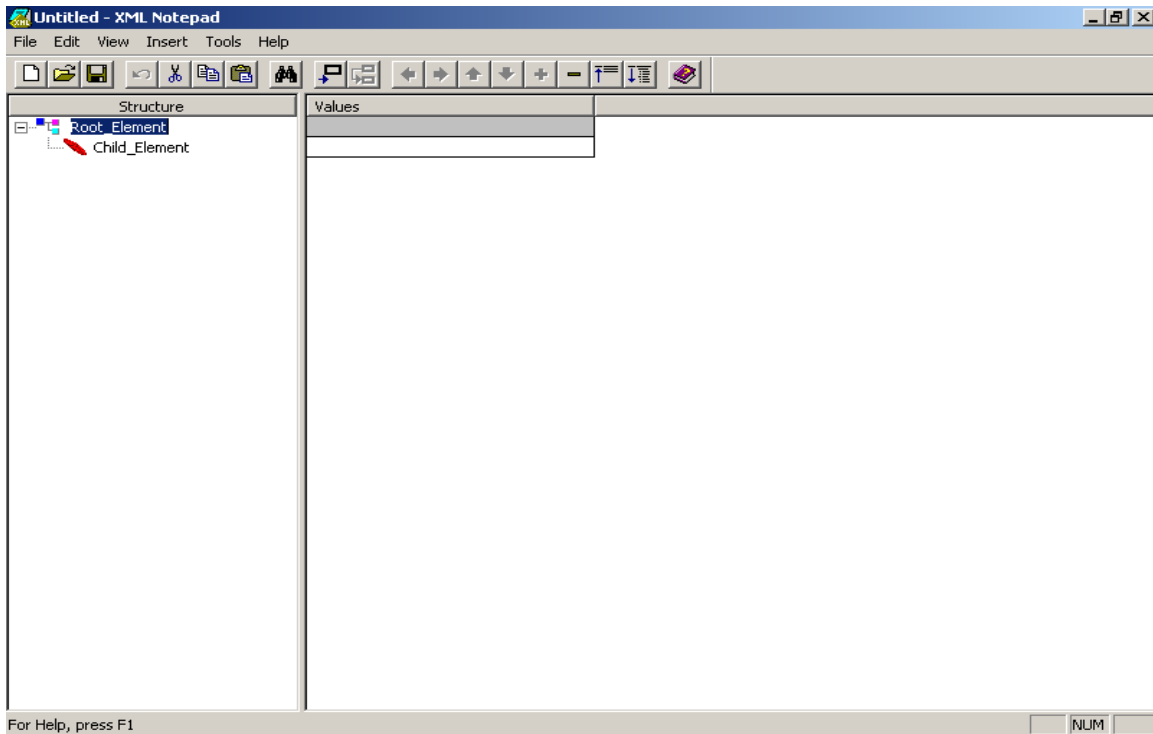


Fig 1 APP-A : XMLNotepad Work Area

For writing XML documents using XML NotePad. You can open a new document and provide information in respective text boxes in the left hand side pane and create XML elements. In the right hand side pane you can provide values for elements created.

The screenshot below displays a sample XML document which is created using XML NotePad.

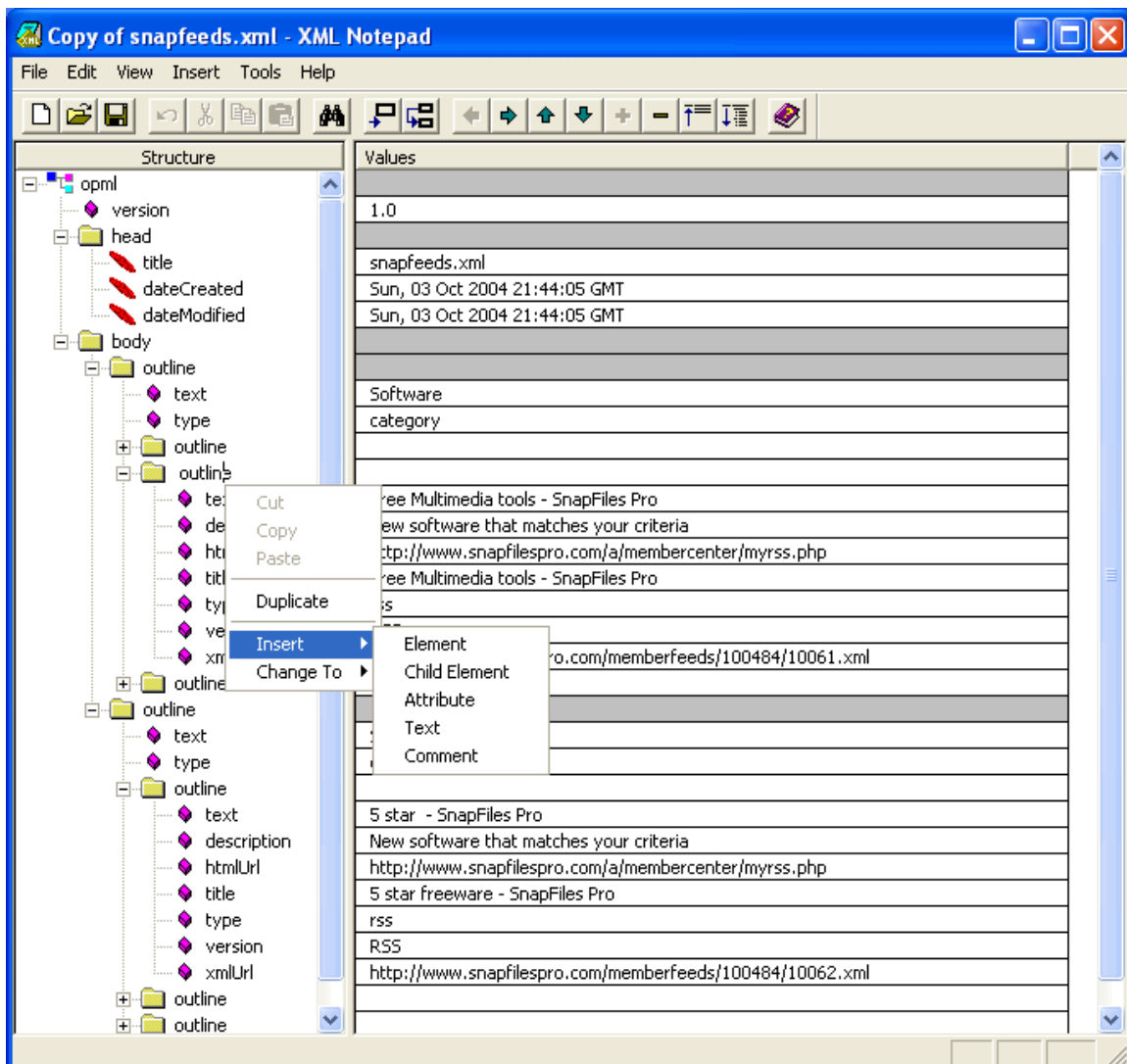


Fig 2 APP-A : Creating XML documents using XML notepad

2. **XML Spy** : The eXtensible Markup Language (XML) has received widespread support and adoption in the computer industry, and XML technologies now play a critical role in all software development projects. Today's developer needs a tool for creating, editing, validating, and debugging XML technologies in an efficient, standards-based manner. Altova XMLSpy® 2005 delivers all the power you need to create the most advanced XML applications, yet at the same time it's flexible enough to allow you to work with XML using the views and options that best suit your

business needs and working preferences. XMLSpy® 2005 increases productivity by allowing you to develop higher-quality, standards-conformant XML-based applications more quickly than ever before.

XMLSpy® 2005 abstracts away the complexity of working with XML-based technologies through its intuitive user interface and rich variety of editing views and options. Five synchronized editing views allow you to work with any XML technology in a way that best suits the complexity of the document and your preferences. XMLSpy® 2005 is the only tool on the market that lets you tackle Web services development on all of the major Web services platforms, including Microsoft .NET, J2EE, and Eclipse. Support for integration with Microsoft Visual Studio.NET and Eclipse allows you to seamlessly access the powerful XMLSpy® 2005 features from within these popular development environments.

The XML SPY editor screen will appear as follows:-

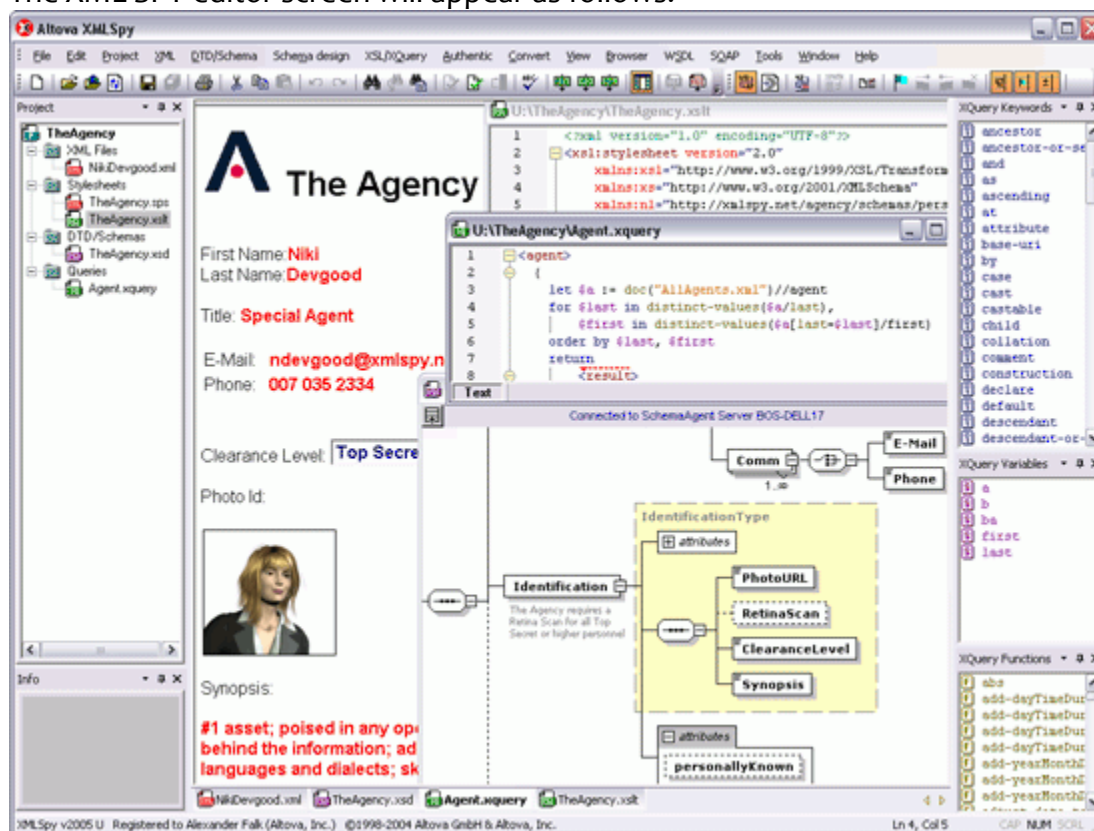


Fig 3 APP-A : XML Spy WorkArea

Appendix B

Validating XML documents against DTD or Schemas.

When we create a XML document we need to check for it a Validity and its WellFormedness. XML Notepad allows us to only check the Wellformednes of the

XML LabBook

XML document. But checking for Well formedness is not enough and we need to validate the XML document against a DTD or Schema. To validate XML documents for our discussion we will use XML Spy. But there are many other XML parsers available which will allow you to check the validity and Well formedness of the document.

1. **Validating a XML document against a DTD:** Open the DTD file and the XML file. In this example we are considering an external DTD. The files are addressbook.dtd and addressbook.xml.

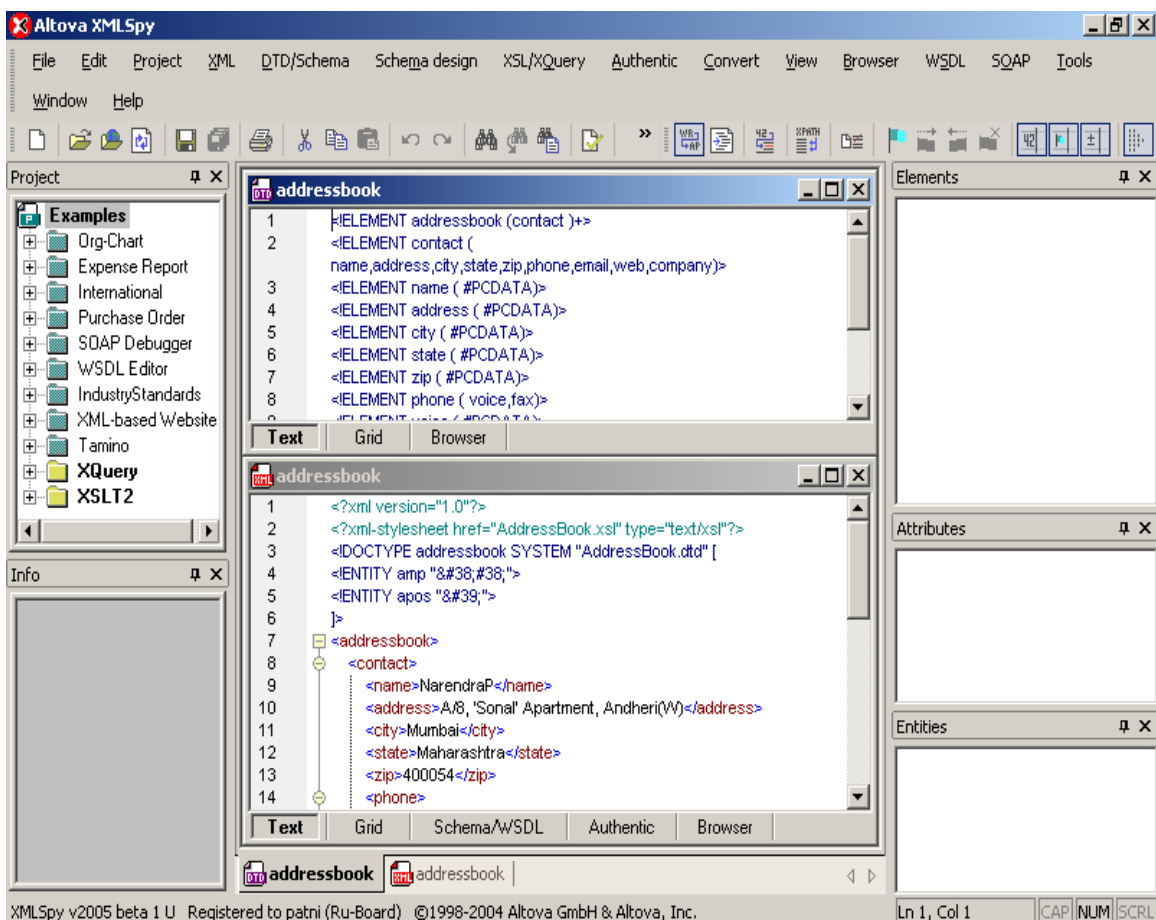


Fig 1 APP-B : DTD and XML document opened in XML spy

Even if you do not open the DTD file it is fine. But make sure that the DTD exist in the appropriate path. After we have opened both the files. Select the Validate option from the XML menu.

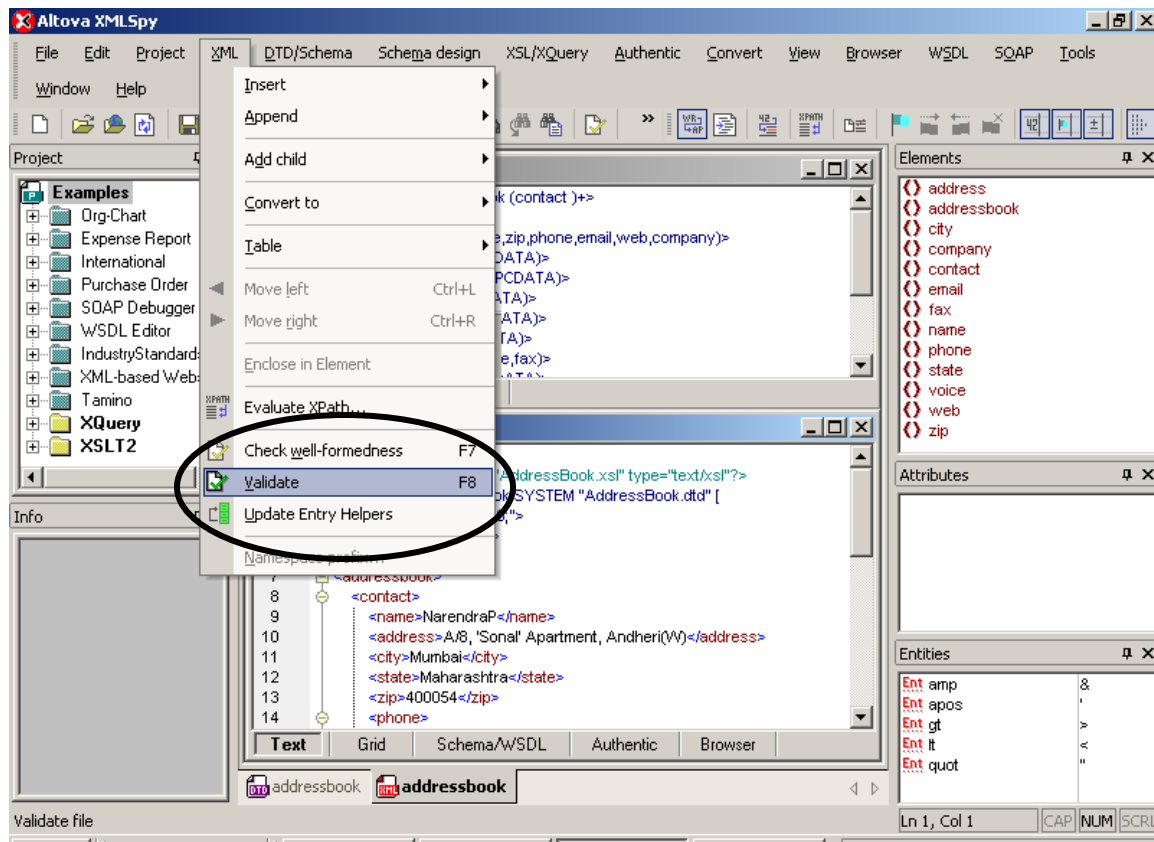


Fig 2 APP-B: Validate menu selected.

After you click on the validate XML Spy will validate your XML document against the DTD. If the file is valid, the output will appear as follows:-

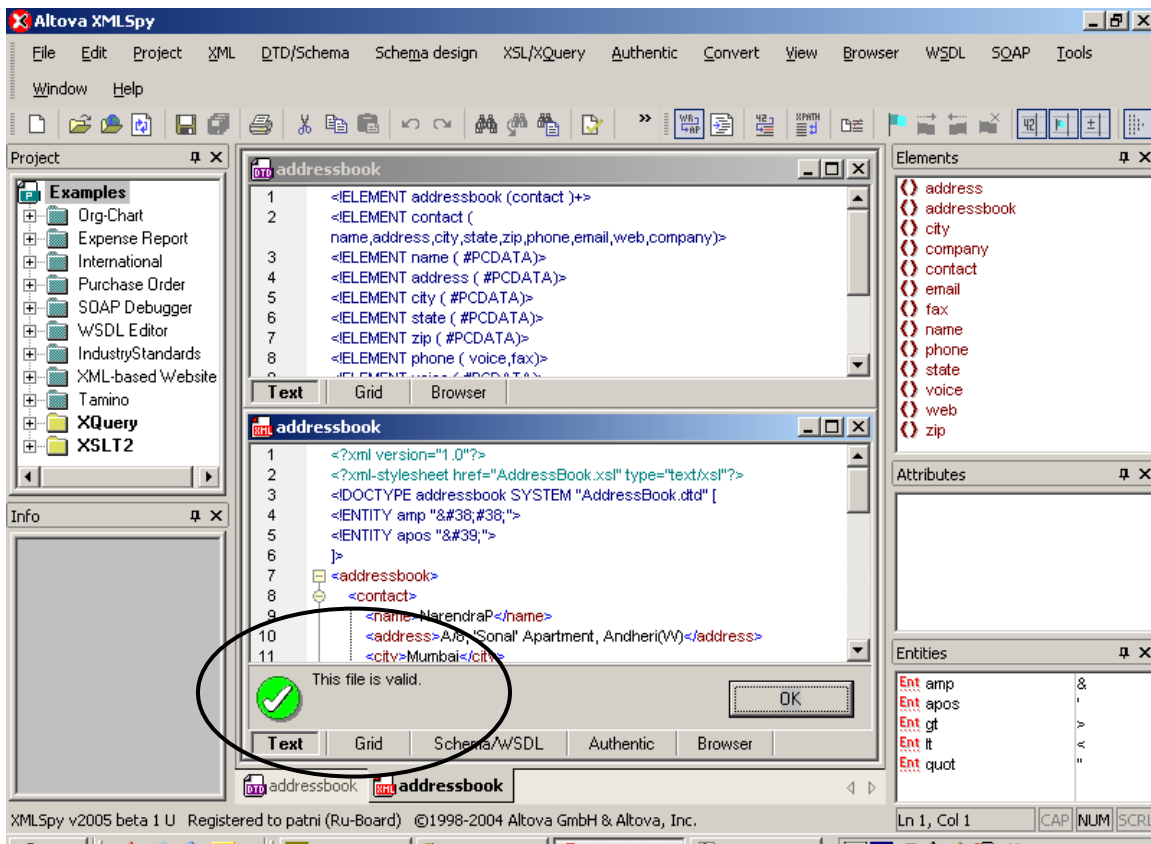


Fig 3 APP-B: Valid XML file against DTD

In case the file is invalid i.e the XML document does not meet the specifications of DTD , XML Spy terms the XML document as invalid.

The output appears as follows and it also indicates the element due to which the file is invalid:-

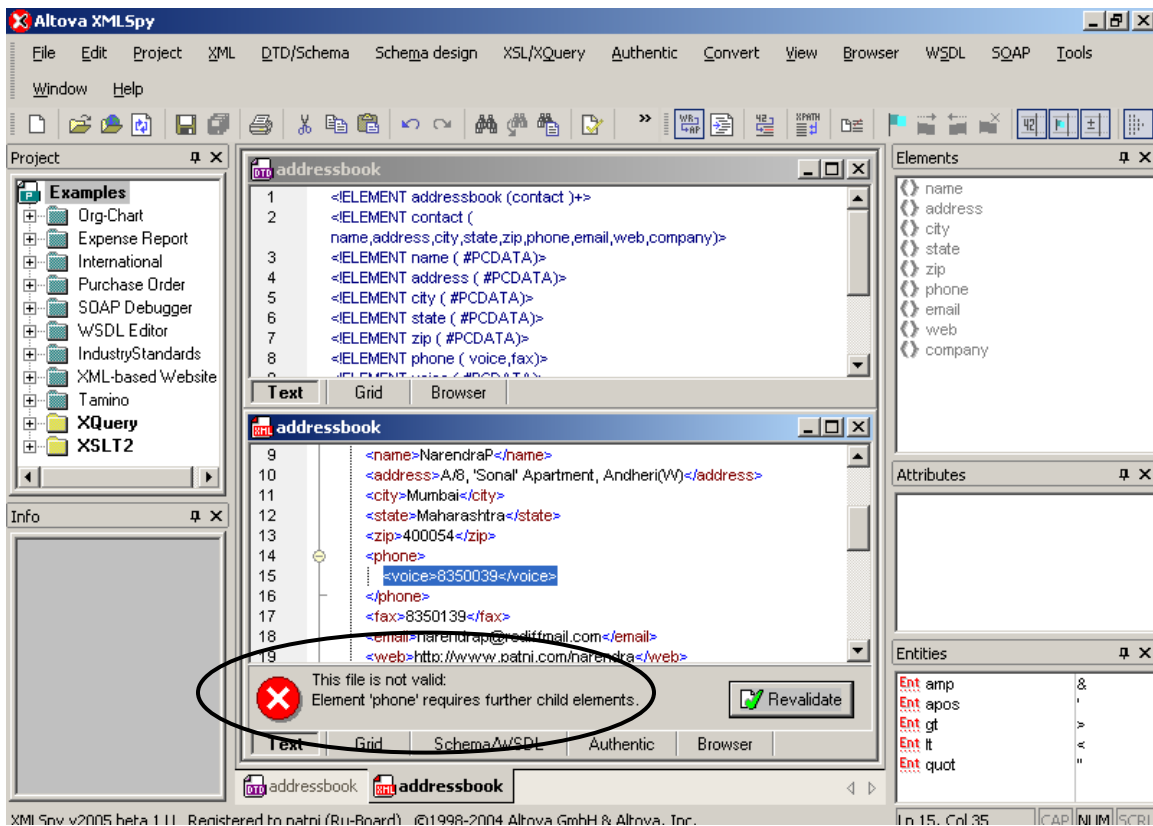


Fig 4 APP-B: Invalid XML file against DTD

2. **Validating a XML document against a Schema :** Open the XSD file and the XML document . The files that we are considering in this example are shiporder.xsd and shiporder.xml.

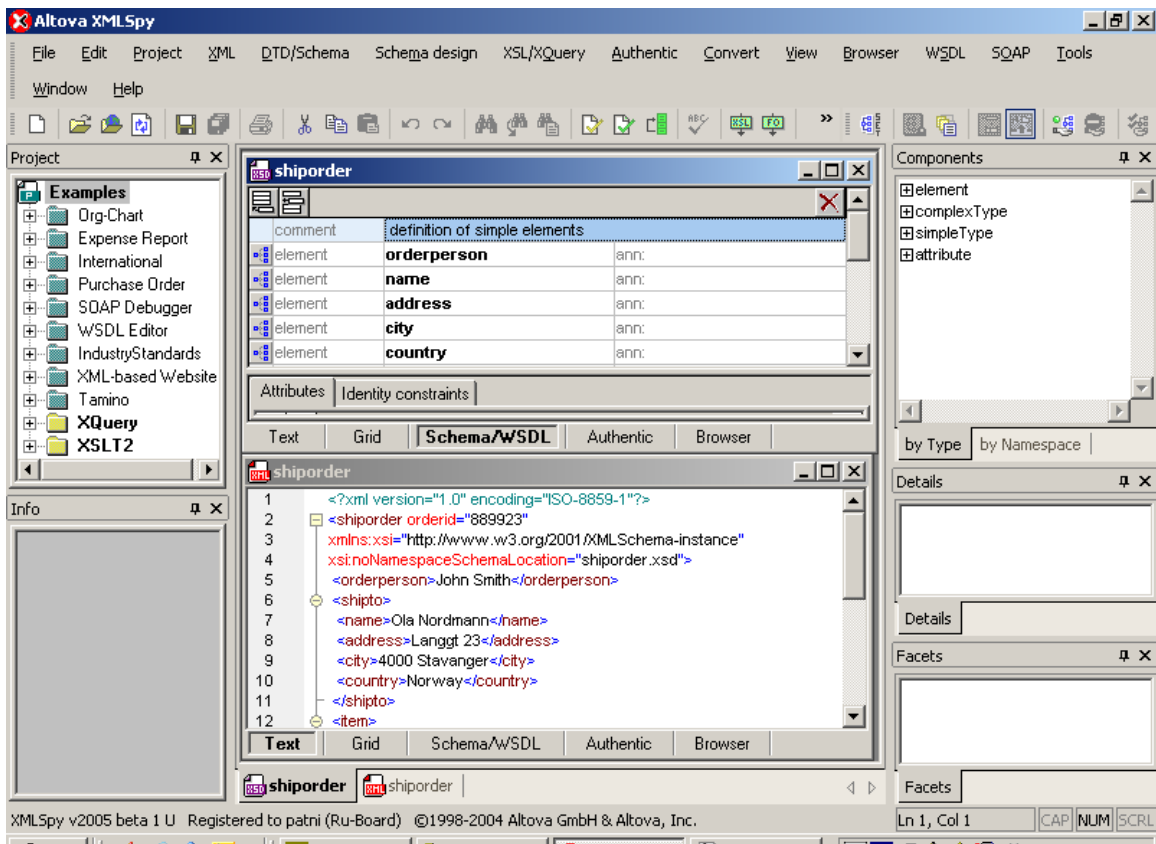


Fig 5 APP-B: XML file and XSD file opened in XML Spy

Even if you do not open the XSD file it is fine. But make sure that the XSD exist in the appropriate path. After we have opened both the files. Select the Validate option from the XML menu.

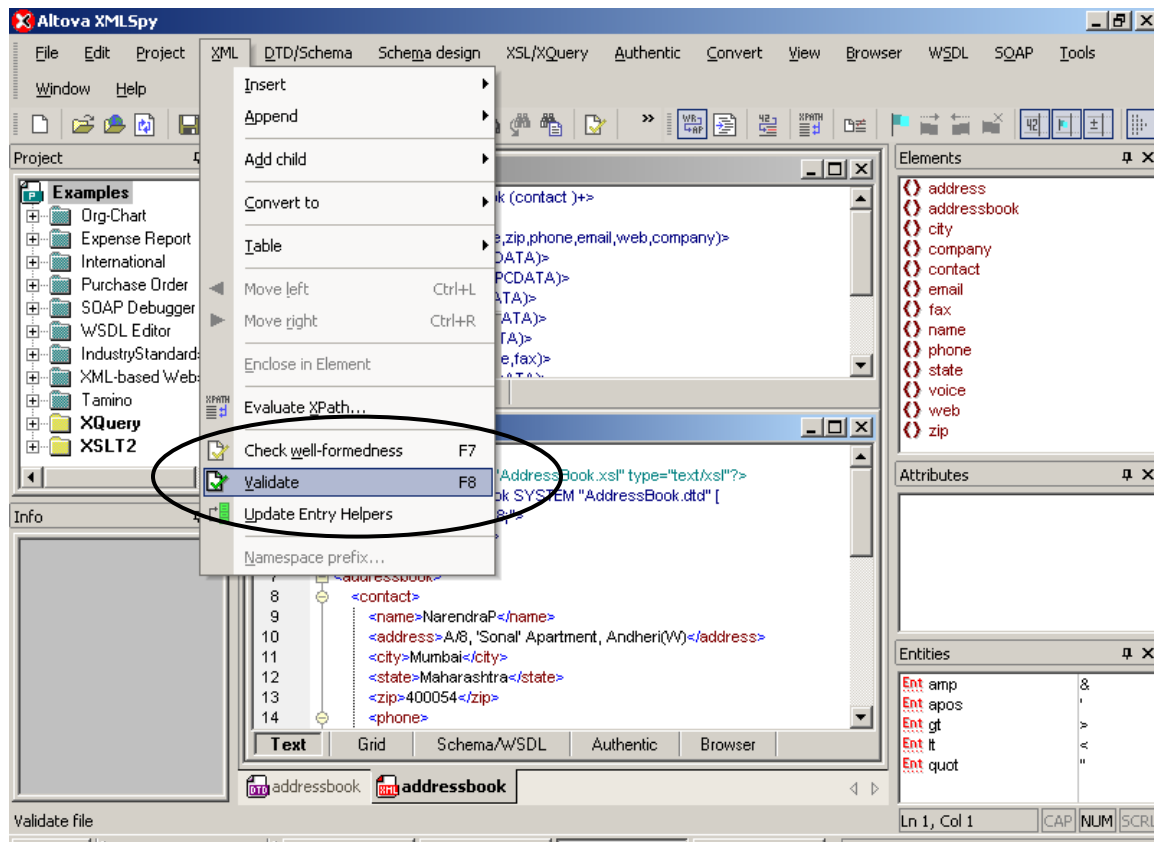


Fig 6 APP-B: Validate menu selected

After you click on the validate XML Spy will validate your XML document against the XSD. If the file is valid, the output will appear as follows:-

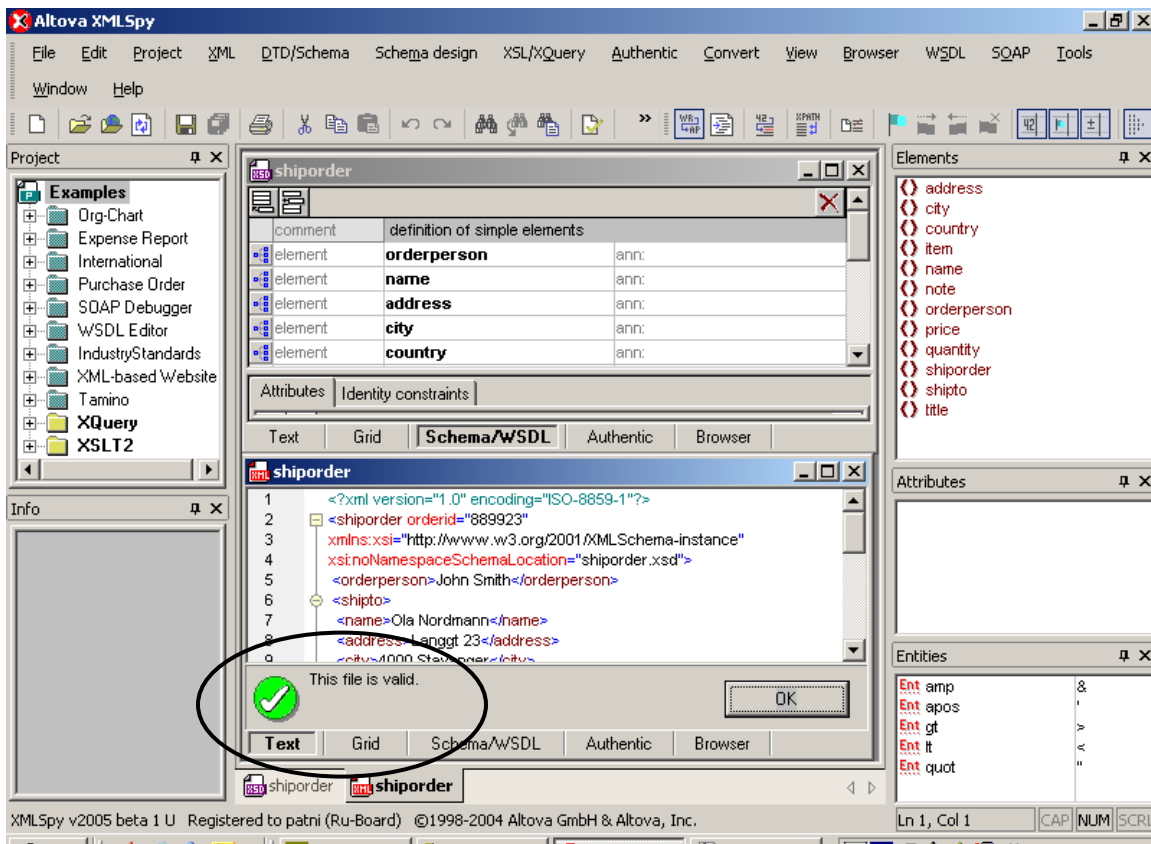


Fig 7 APP-B: Valid XML file against XSD

In case the file is invalid i.e the XML document does not meet the specifications of XSD , XML Spy terms the XML document as invalid.
The output appears as follows and it also indicates the element due to which the file is invalid:-

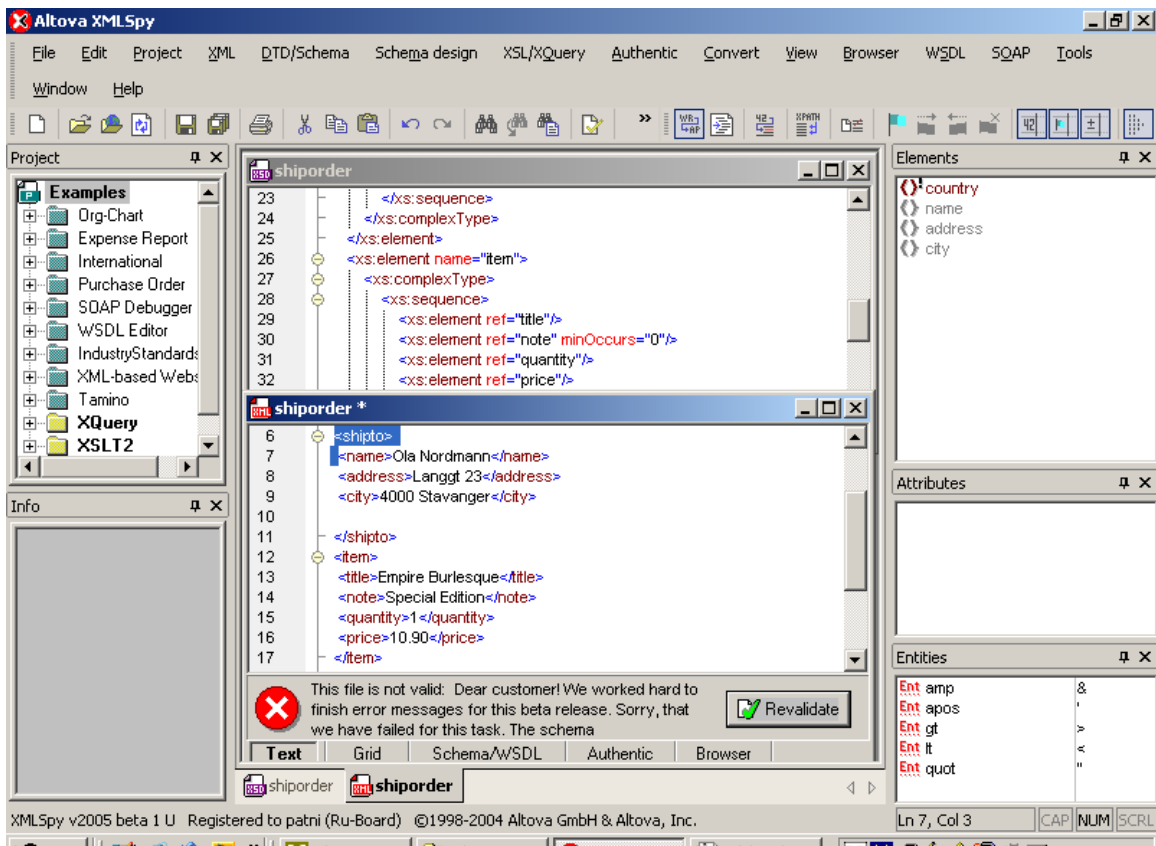


Fig 8 APP-B: Invalid XML file against XSD.