LookyBook Tutorial

Introduction

This tutorial shows how to use the “LookyBook” web service found at
http://www.xmethods.com/ve2/ViewListing.po?serviceid=94452
The description of this web service is:

This web service consists of a single method, GetInfo, with a single
input param: ISBN (International Standard Book Number).

The web service retrieves the name, title, publisher, author, and
other data about the book. Also retrieves price data from Amazon,
Barnes & Noble, and bookshop.co.uk.

We will build a form that allows a user to enter an ISBN, and shows the information returned in a form. The screen shots are based on Delphi 5, but you can use any supported version of Delphi or Kylix for this.

Requirements

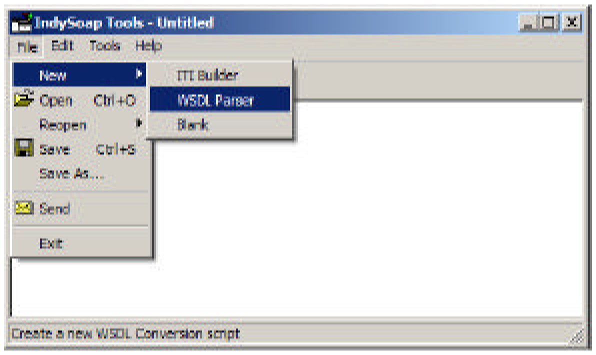
You need to have a version of IndySoap 0.02 or more recent installed.

Overview

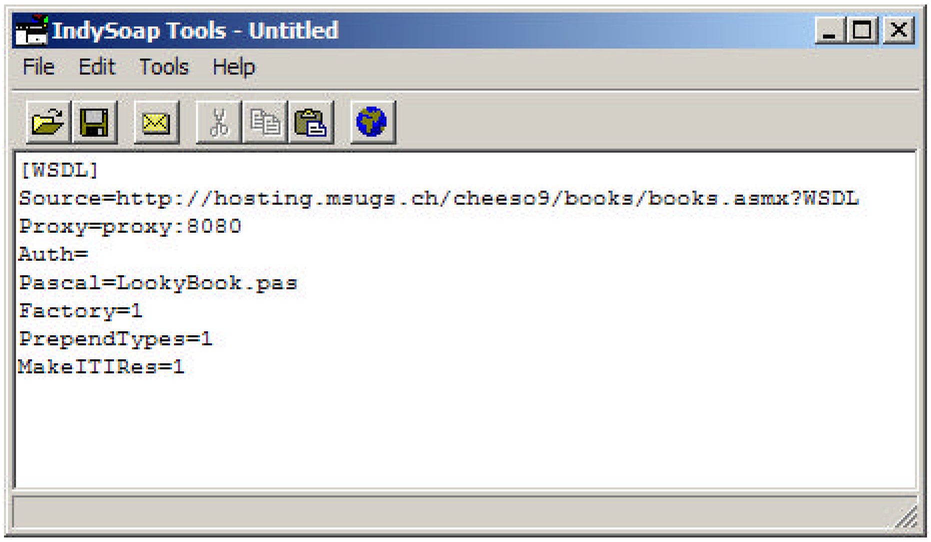
The process for using this webservice consists of generating pascal source and other metadata from the WSDL using the IndySoap Tools, then writing code that uses the interface generated by the Tools.

Generation of Pascal Source and Metadata

1. Create a work directory for the project. This example will use c:\temp\lookybook
2. Start up the IndySoap Tools. Create a new WSDL Parser



3. From the URL above, the WSDL source is http://hosting.msugs.ch/cheeso9/books/books.asmx?WSDL
4. Enter your proxy details as appropriate
5. The metadata is known as “ITI” (Interface Type Information). You can produce this a native file or a resource that will be compiled straight into the application. We will use a resource
6. Fill out other details as shown



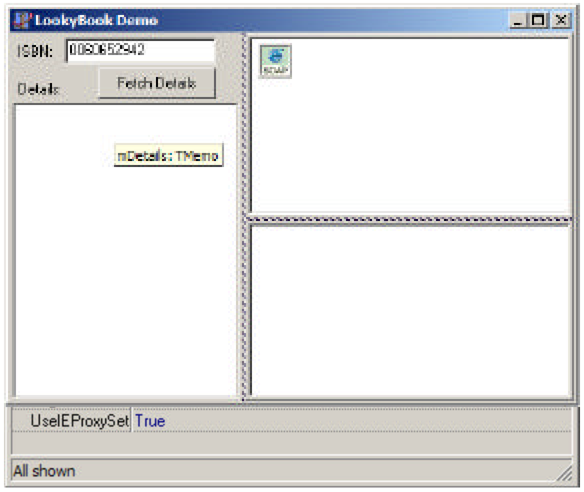
```
[WSDL]
Source=http://hosting.msugs.ch/cheeso9/books/books.asmx?WSDL
Proxy=proxy:8080
Auth=
Pascal=LookyBook.pas
Factory=1
PrependTypes=1
MakeITIRes=1
```

7. Save this file as LookyBook.IdSoapCfg in the work directory
8. Execute (tools menu). You should get 2 files, Lookybook.pas and LookyBook.res

Using the LookyBook Interface in a Pascal Program

1. Create a new application. Save it in the working directory
2. On the form, place a TIdSoapClientWinInet object, and set the SoapURL, ITISource, ITIResourceName, and active as shown:

   The resource name is “LookyBook” because by default the name of resource is the same as the name of the resource file (which is included in LookyBook.pas)



3. On the form, place 3 memos, and edit box, and a button
4. Add the Unit LookyBook to the implementation uses clause
5. Add the following code to the Button OnClick event to get an instance of the interface for the web service:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  LIntf : ILookyBook;
begin
  LIntf := GetILookyBook(IdSoapClientWinInet1);
end;
```

6. To use the LookyBook interface, you need to look in the LookyBook Unit. There you will see that it has a single method that takes a ISBN as a string, and returns a TbookInfo object.
7. We can do this in pascal as follows:|

```
procedure TForm1.Button1Click(Sender: TObject);
var
  LIntf : IlookyBook;
  LInfo : TbookInfo;
begin
  LIntf := GetILookyBook(IdSoapClientWinInet1);
  LInfo := Lintf.GetInfo(eISBN.text);
end;
```

8. Then we simply iterate the TbookInfo structure to populate the memo field:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  LIntf : ILookyBook;
  LInfo : TbookInfo;
  i : integer;
  s : string;
begin
  LIntf := GetILookyBook(IdSoapClientWinInet1);
  LInfo := LIntf.GetInfo(eISBN.text);
  if assigned(LInfo) then
    begin
    try
      s :=
        'isbn: '+LInfo.isbn+crlf+
        'title: '+LInfo.title+crlf+
        'author: '+LInfo.author+crlf+
        'pubdate: '+LInfo.pubdate+crlf+
        'publisher: '+LInfo.publisher+crlf+
        'imgUrl: '+LInfo.imgUrl+crlf+
        'timestamp: '+LInfo.timestamp+crlf;
      for i := Low(LInfo.vendorprice) to
                High(LInfo.vendorprice) do
        begin
        s := s + '  '+Linfo.vendorprice[i].name+
               ':'+Linfo.vendorprice[i].pricePrefix+
               Linfo.vendorprice[i].price+crlf;
        end;
      mDetails.text := s;
    finally
      FreeAndNil(LInfo);
    end;
    end
  else
    begin
    showmessage('no book found');
    end;
end;
```

9. Comments about this code:
   - What isn't described in the WSDL, but quickly becomes evident on testing is that the routine will return result = nil in the case of error).
   - You need to free the LInfo object yourself manually. You can set the property GarbageCollectObjects on the TIdSoapClientWinInet to true, and then you won't have to. It's a matter of personal preference whether you want objects garbage collected, or you wish to maintain control over them
10. You should be able to execute this, and it will work. The WinInet based client will automatically pick up your proxy settings
11. For interest, you can see the XML associated with the SOAP calls. Add IdSoapUtilities to the uses clause, and the following code to the OnSendMessage and OnReceiveMessage events:

```
        procedure TForm1.IdSoapClientWinInet1SendMessage
                     (ASender: TObject; AMessage: TStream);
        var
          s : string;
        begin
          SetLength(s, AMessage.Size);
          AMessage.Read(s[1], AMessage.Size);
          mSent.text := IdSoapMakeXmlPretty(s);
        end;

        procedure TForm1.IdSoapClientWinInet1ReceiveMessage
                  (ASender: TObject;  AMessage: TStream);
        var
          s : string;
        begin
          SetLength(s, AMessage.Size);
          AMessage.Read(s[1], AMessage.Size);
          mRecvd.text := IdSoapMakeXmlPretty(s);
```

```
              end;
```

12. Run again, you will see the XML. You can also transform the XML etc at this point.

The source for this tutorial is in the IndySoap tutorial directory as LookyBook.zip