

Using the SOA Gateway to access the Sybase database

Using the SOA Gateway to access the Sybase EAServer database Adaptive Server Anywhere.

- Introduction
 - ODBC Setup
 - SOA Gateway Configuration
 - Accessing SQL Server using the SOA Gateway
-

Introduction

The EAServer product set from Sybase includes Adaptive Server Anywhere. Adaptive Server Anywhere provides a full-featured SQL database for EAServer applications. It is designed to operate in environments with limited physical and database administration resources. Adaptive Server Anywhere is a transaction-processing relational database management system (RDBMS) with full recovery capabilities, online backup, referential integrity actions, stored procedures, triggers, row-level concurrency control, and a rich SQL language.

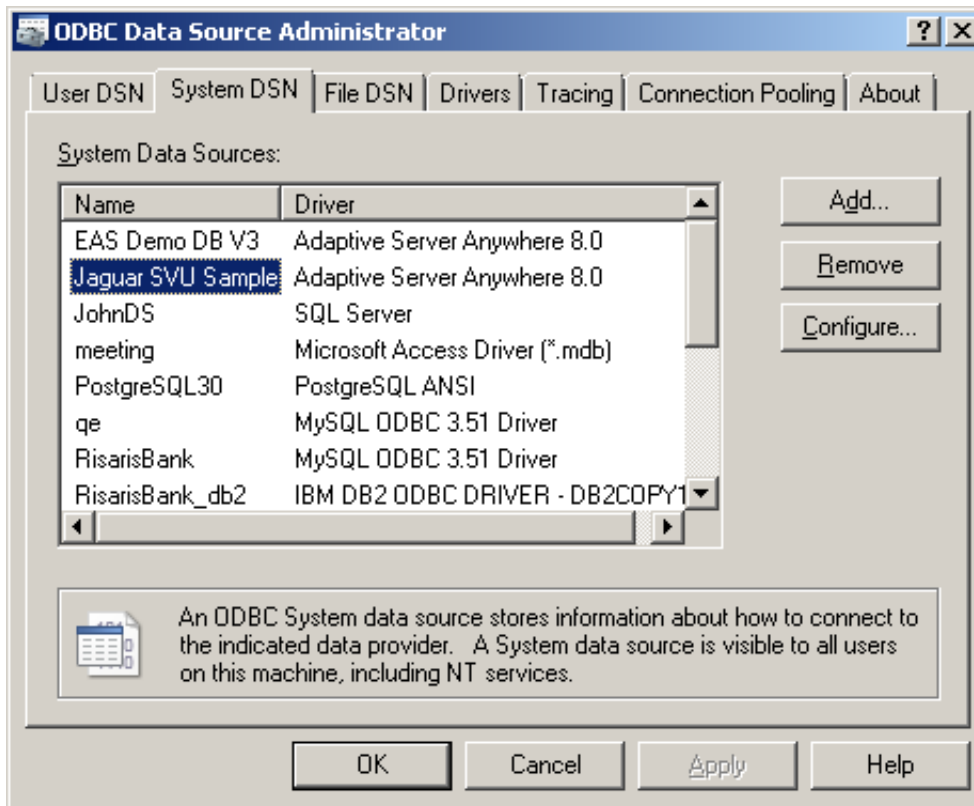
Go to [here](#) to download the Sybase EAServer Developer Edition which is used in this example.

This HOWTO assumes that the SOA Gateway and the Sybase database are both running on Windows.

For the purpose of this HOWTO, we will be using one of the sample databases provided by Sybase, "Jaguar SVU Sample".

ODBC Setup

By default, the Sybase installation sets the DSN information required for accessing the sample database:



Check that the login details are correct.

Select the Configure... button having first highlighted the sample database as shown above.

Select the Login tab:

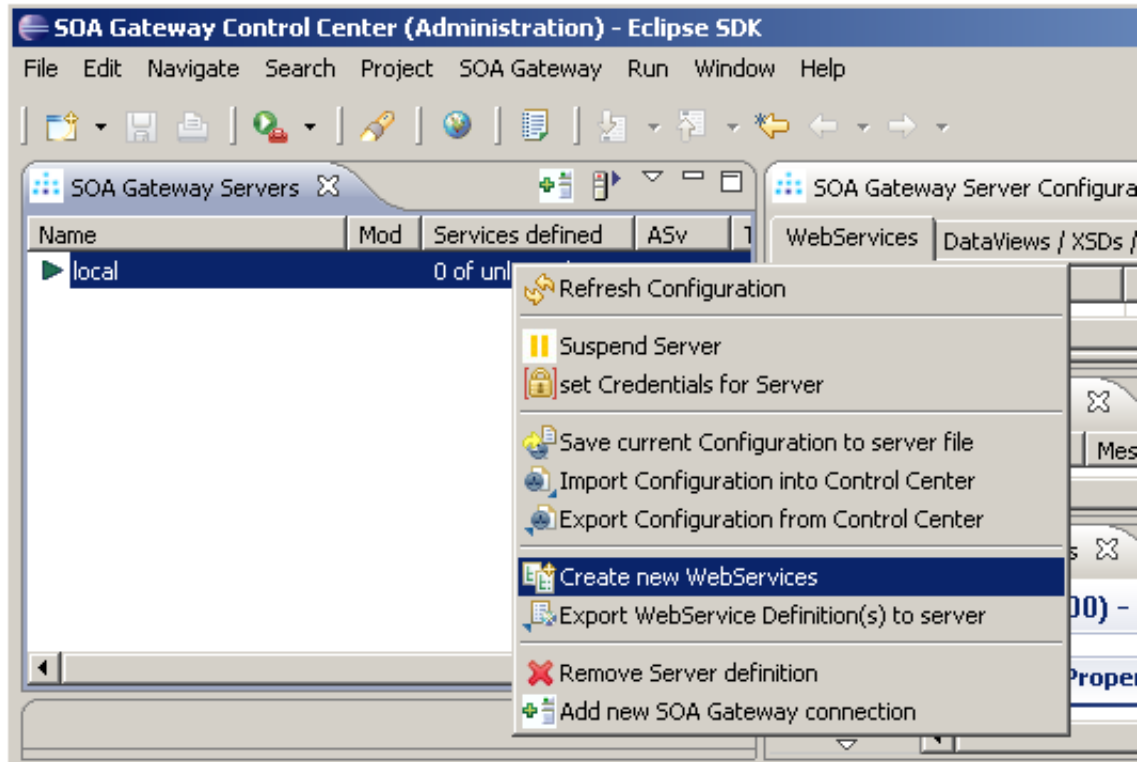


If the User ID and Password is not set set them to dba and sql respectively.

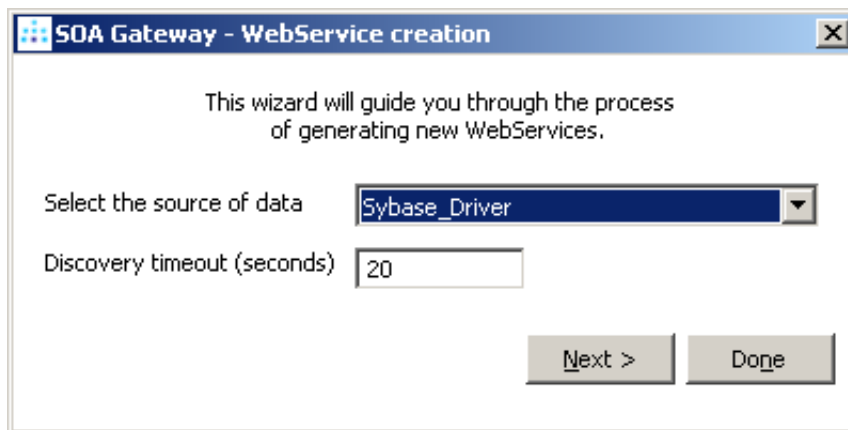
SOA Gateway Configuration

The SOA Gateway must now be configured to access and use this DSN

- Start the SOA Gateway Control Center and add a SOA Gateway Server (if necessary). See here for more information.
- If you do not have an Sybaser driver, add one now. See here for more information.
- Right click on your server, and select **Create new WebServices:**



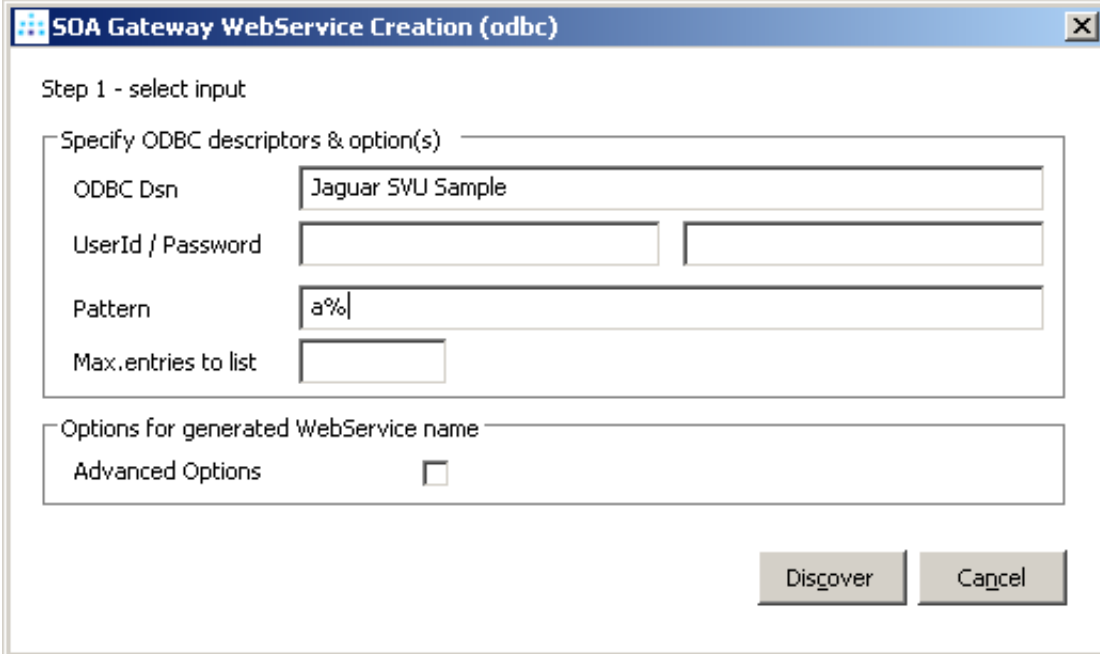
- From the next dialog choose **Sybase_Driver** and click **Next**



- The next dialog prompts you for the the
 1. **ODBC Dsn:** In this case it will be Jaguar SVU Sample
 2. **UserId / Password:** As it has been set at DSN level this is not necessary. Otherwise set as appropriate.
 3. **Pattern:** This will be a pattern match value passed to the SQL Command. The search pattern characters are: an underscore (_), which represents any single character and a percent sign (%), which represents any sequence of zero or more characters.
 4. **Max. entries to list:** This is a hard-limit on the amount of tables the SOA Gateway will attempt to discover.

5. Advanced Options: Enables changing the generated WebService name. Leave unset.

Example:



SOA Gateway WebService Creation (odbc)

Step 1 - select input

Specify ODBC descriptors & option(s)

ODBC Dsn Jaguar SVU Sample

UserId / Password

Pattern a%

Max. entries to list

Options for generated WebService name

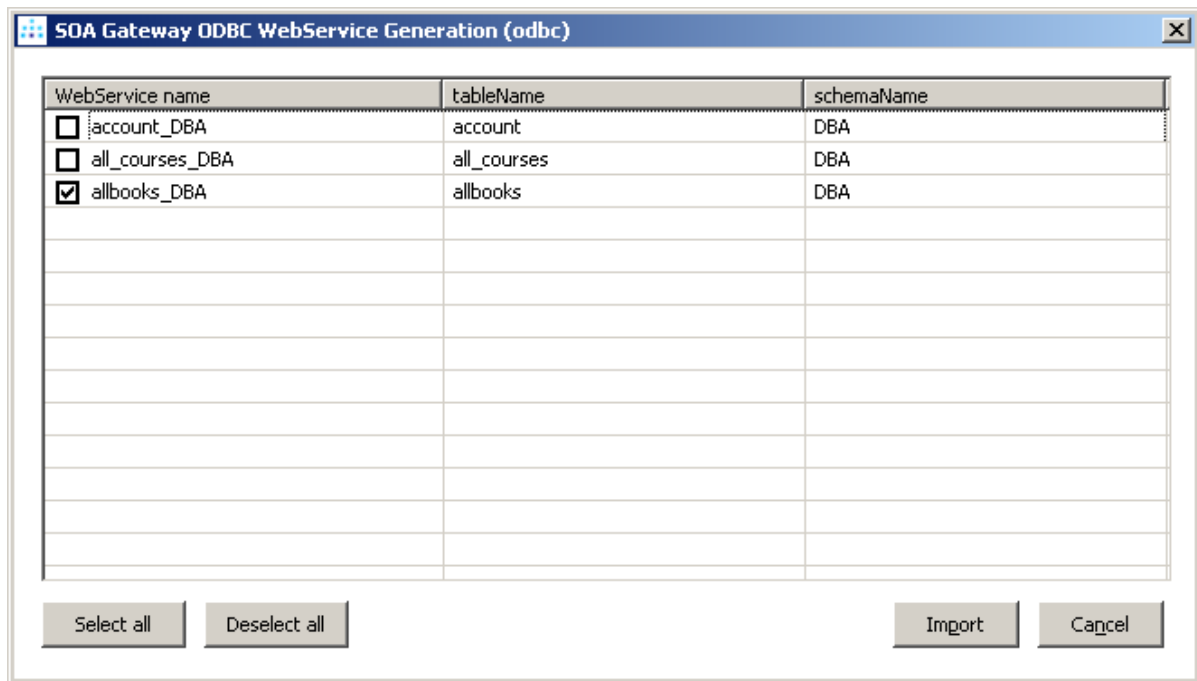
Advanced Options

Discover Cancel

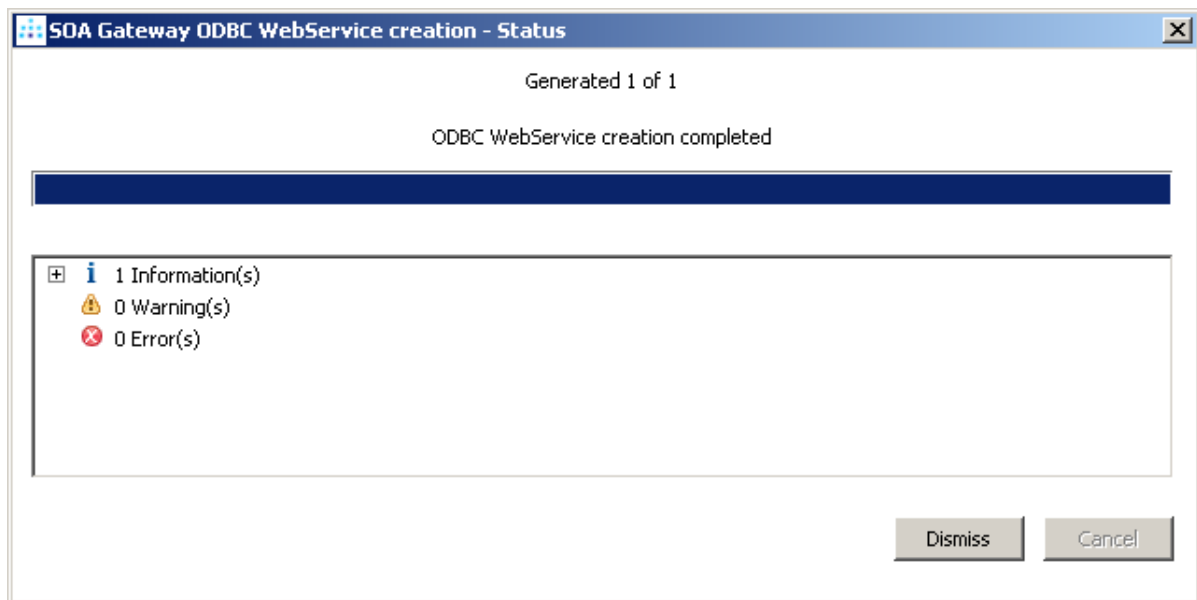
Now click **Discover**

- The SOA Gateway will ask the Sybase Server database (identified by the Jaguar SVU Sample DSN) to display all the tables which match the request. In our case, the number of tables starting with 'a' were returned.

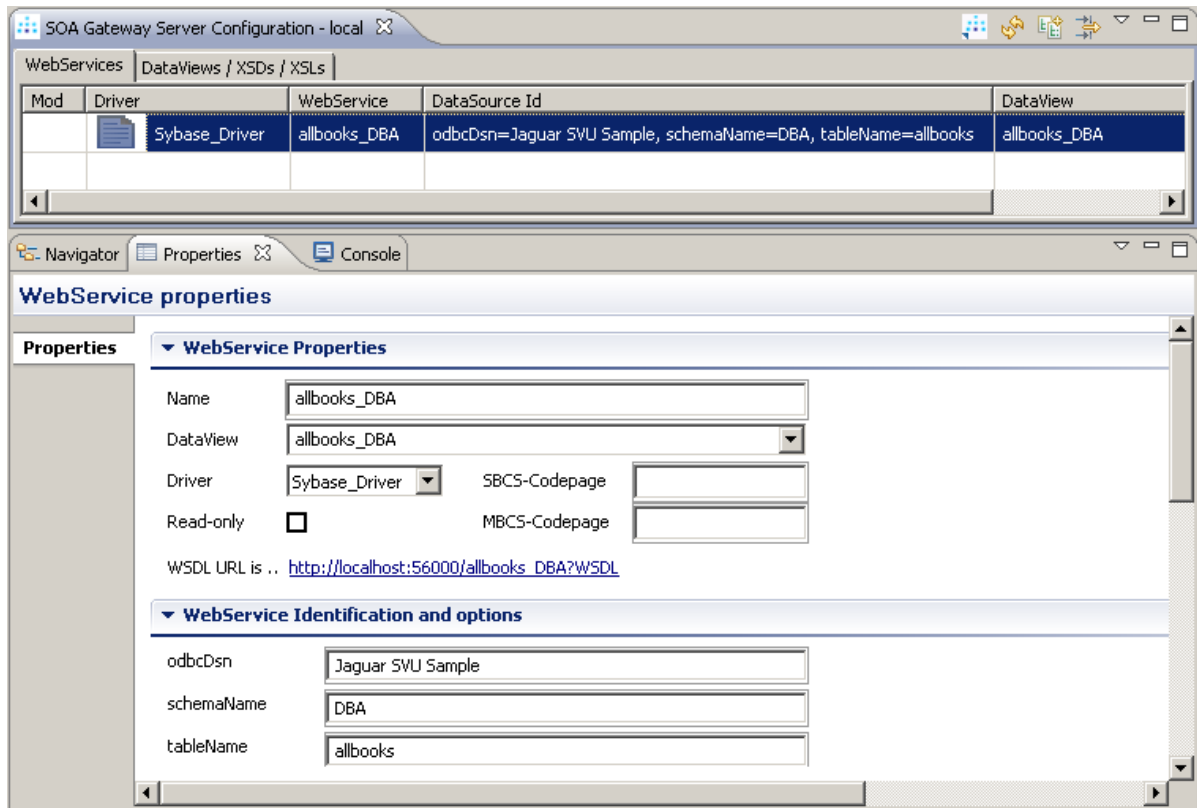
Select table allbooks, and click **Import**



- The results of the import will be displayed in a dialog



- Click **Dismiss** to finish the Web Service creation
- You have now created a Web Services based on table allbooks from the Sybase database!



Accessing SQL Server using the SOA Gateway

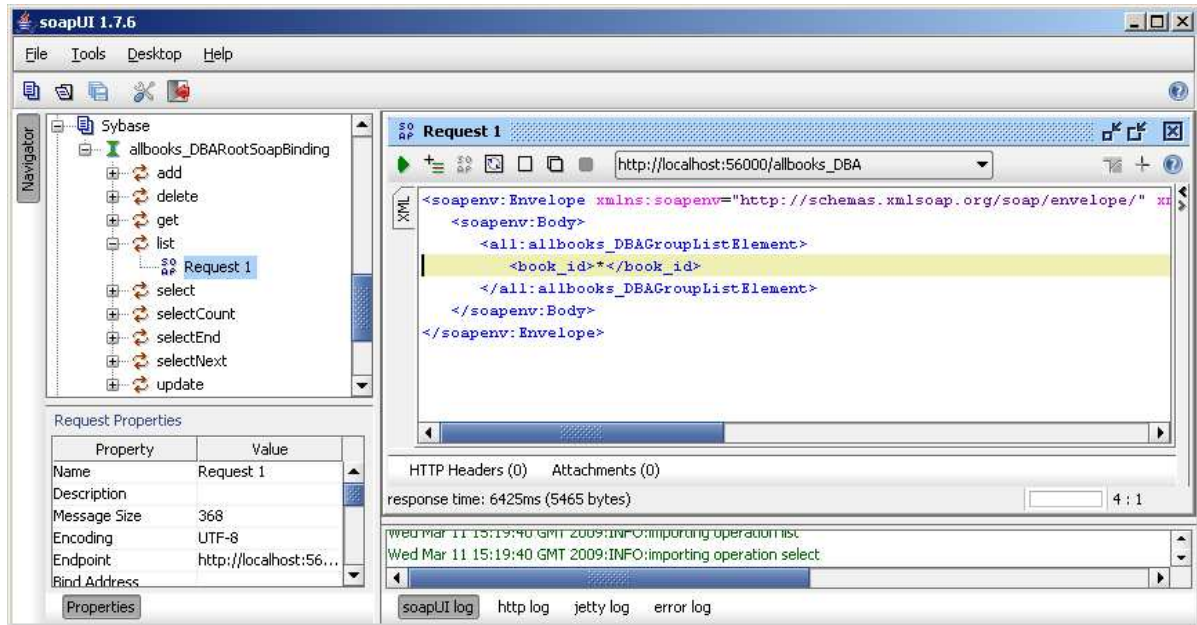
Now that the Web Services have been set up, you can access the Web Service Description Language (WSDL) by clicking in the WSDL URL link in the Control Centre.

This WSDL is the starting point to accessing the your tables and stored procedures using the SOA Gateway. There are many clients available to consume and use web services, for example soapUI, XMLSpy, and InfoPath.

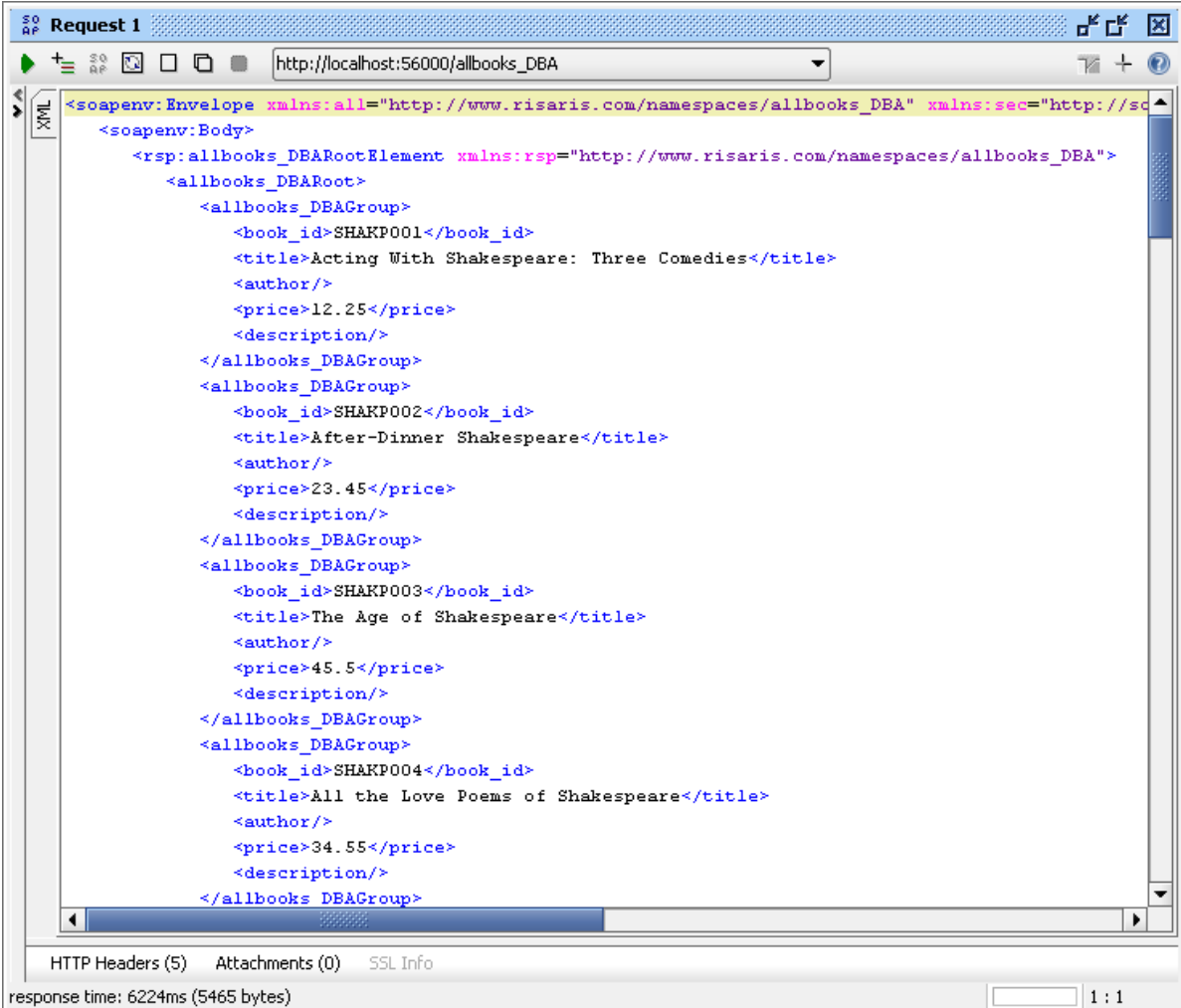
A tutorial on how to access Adabas data through soapUI is available here. The follow example is based on that tutorial, and shows how to access the allbooks table we have just enabled.

- Start soapUI and create a new WSDL project.
- Import the allbooks WSDL, E.g. http://localhost:56000/allbooks_DBA?WSDL
- Edit the list request
- Completely remove the <all:allbooks_DBAGroupHeader> element from the <soap:Header> element
- In the <Security> element, add Username and Password for accessing the allbooks table if required. Otherwise, the <Security> element, and indeed the rest of the <Header> element can be removed as shown below.
- Add "*" as the content of the <book_id> element (same as `SELECT * from allbooks;`)

E.g



- Hit the green arrow, and the results of the request should be displayed



The screenshot displays a SOAP client window titled "Request 1" with the URL `http://localhost:56000/allbooks_DBA`. The main content area shows the XML response body, which is a SOAP envelope containing a list of books. The XML structure is as follows:

```
<soapenv:Envelope xmlns:all="http://www.risaris.com/namespaces/allbooks_DBA" xmlns:sec="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <rsp:allbooks_DBARootElement xmlns:rsp="http://www.risaris.com/namespaces/allbooks_DBA">
      <allbooks_DBARoot>
        <allbooks_DBAGroup>
          <book_id>SHAKP001</book_id>
          <title>Acting With Shakespeare: Three Comedies</title>
          <author/>
          <price>12.25</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP002</book_id>
          <title>After-Dinner Shakespeare</title>
          <author/>
          <price>23.45</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP003</book_id>
          <title>The Age of Shakespeare</title>
          <author/>
          <price>45.5</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP004</book_id>
          <title>All the Love Poems of Shakespeare</title>
          <author/>
          <price>34.55</price>
          <description/>
        </allbooks_DBAGroup>
      </allbooks_DBARoot>
    </rsp:allbooks_DBARootElement>
  </soapenv:Body>
</soapenv:Envelope>
```

At the bottom of the window, there are tabs for "HTTP Headers (5)", "Attachments (0)", and "SSL Info". The status bar at the bottom left indicates "response time: 6224ms (5465 bytes)" and the bottom right shows "1 : 1".

Congratulations! You have now accessed a Sybase table using the SOA Gateway!