Deltek VisionXtend™ 7.4

Testing the Vision Web APIs / Web Services

January 23, 2015

This edition published January 2015.

Contents

Overview

Web application programming interfaces (Web APIs) or Web services that access Vision Info Centers are used by external applications that are created to make calls to the APIs. The development of these external applications requires understanding of how to send data to and also receive (and interpret) messages from Vision Web APIs. To facilitate testing of Web APIs for Vision, Deltek recommends that you use soapUI. soapUI is free, third-party software that you can use to run simple tests for the Vision APIs that you create. For example, you can call individual API methods and script multiple calls with soapUI.

Use soapUI with APIs only for preliminary testing purposes. Deltek does **not** recommend that you use it in an expanded capacity. The examples in this document do not cover the full scope of using or trying APIs in all scenarios.

Adding Custom Notes to This Guide

If you would like to add custom notes to this guide that are specific to your company, Adobe® Reader® X provides this ability. If you do not already use Adobe Reader X, you can download it here free from Adobe.

To add a custom note using Adobe Reader X, complete the following steps:

1. On the Reader toolbar, click **Comment** at the far right.
2. In the **Annotations** pane that displays, click **Sticky Note**. The cursor changes to match the button.
3. Position the cursor at the location in the guide where you want the note to appear, and click. A note icon is inserted at the location and a text box pops up.
4. Enter your information in the text box.
5. Continue to add notes as needed.
6. Save the document.

Deltek recommends that you save the document to a slightly different filename so as to keep the original file from being overwritten.

When you read the document, cursor over a note icon to see the information. Double-click a note icon to edit the information.

If You Need Assistance

If you need assistance installing, implementing, or using Vision, Deltek makes a wealth of information and expertise readily available to you.

Customer Services

For over 20 years, Deltek has maintained close relationships with client firms, helping with their problems, listening to their needs, and getting to know their individual business environments. A full range of customer services has grown out of this close contact, including the following:

- Extensive self-support options through the Customer Care Connect Web portal.
- Phone and email support from Customer Care analysts
- Technical services
- Consulting services
- Custom programming
- Classroom, on-site, and Web-based training

Find out more about these and other services from the Customer Care Connect site.

Customer Care Connect Site

The Deltek Customer Care Connect site is a support Web portal for Deltek customers who purchase an Ongoing Support Plan (OSP).

The following are some of the many options you have at the Customer Care Connect site:

- Download the latest versions of your Deltek products
- Search Deltek's knowledge base
- Ask questions, exchange ideas, and share knowledge with other Deltek customers through the Deltek Connect Customer Forums
- Display or download product information, such as release notes, user guides, technical information, and white papers
- Submit a support case and check on its progress
- Transfer requested files to a Customer Care analyst
- Use Quick Chat to submit a question to a Customer Care analyst online
- Subscribe to Deltek communications about your Deltek products and services
- Receive alerts of new Deltek releases and hot fixes

If you need assistance using the Customer Care Connect site, the online help available on the site provides answers for most questions.

Access Customer Care Connect

To access the Customer Care Connect site, complete the following steps:

1. Go to http://support.deltek.com.
2. Enter your Customer Care Connect **Username** and **Password**.
3. Click **Log In**.

If you do not have a username and password for the Customer Care Connect site, contact your firm's Vision administrator.

If you forget your username or password, you can click the **Account Assistance** button on the login screen for help.

Additional VisionXtend Documentation

The following VisionXtend PDF document is available to download from the Deltek Customer Care Connect site:

Deltek VisionXtend Web Services and APIs for Deltek Vision — This guide explains how to use the Deltek VisionXtend platform to integrate Vision with other applications, access web services, implement data validation routines, and establish workflow procedures using the Microsoft .NET Framework. This guide also includes:

- Guidelines for using a Web service with Vision workflows, including sample code snippets
- Guidelines for developing a custom component or an assembly for workflow conditions
- Information on extending data validation business logic for timesheets
- Information on extending data validation business logic for expense reports

Install soapUI

To install soapUI, go to www.soapui.org and download the software. This “Testing the Vision Web APIs/Web Services” guide includes screenshots and basic instructions using the free download of soapUI (version 4.5.1).

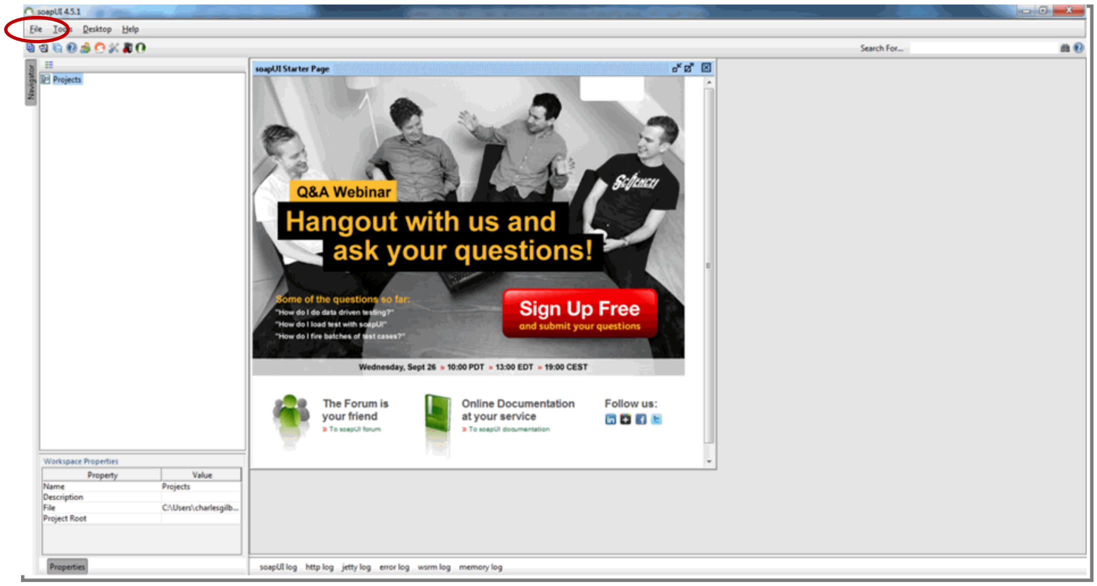
soapUI also comes in a Professional version.

Configure soapUI to Use Your Vision Web Server

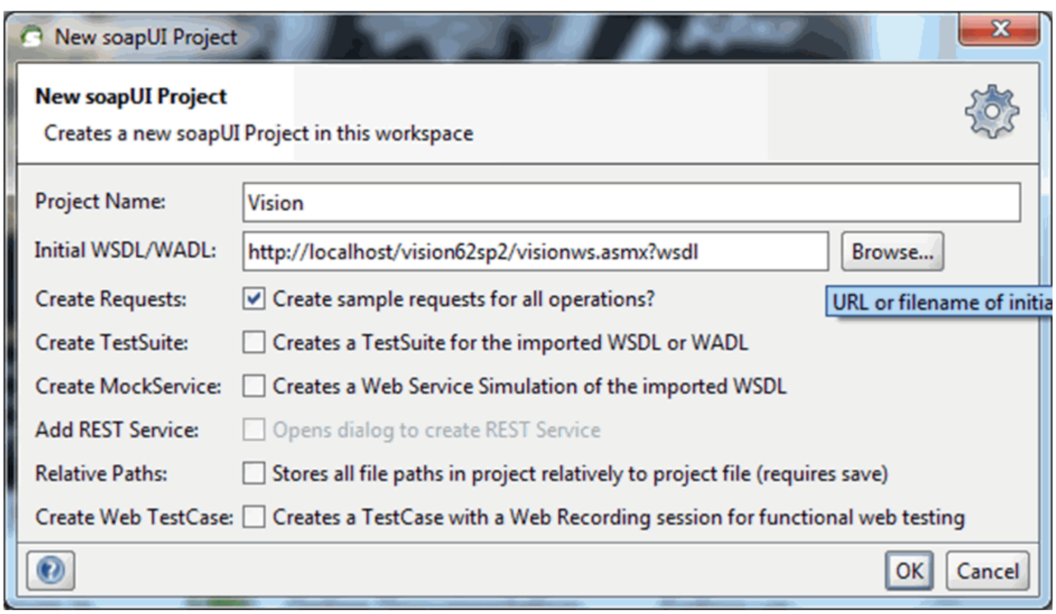
Before you use soapUI with a Vision API, you must configure soapUI to use your Vision web server.

To configure soapUI to use your Vision web server, complete the following steps:

1. Launch soapUI.
2. On the menu bar of the main screen, click **File » New Project**.

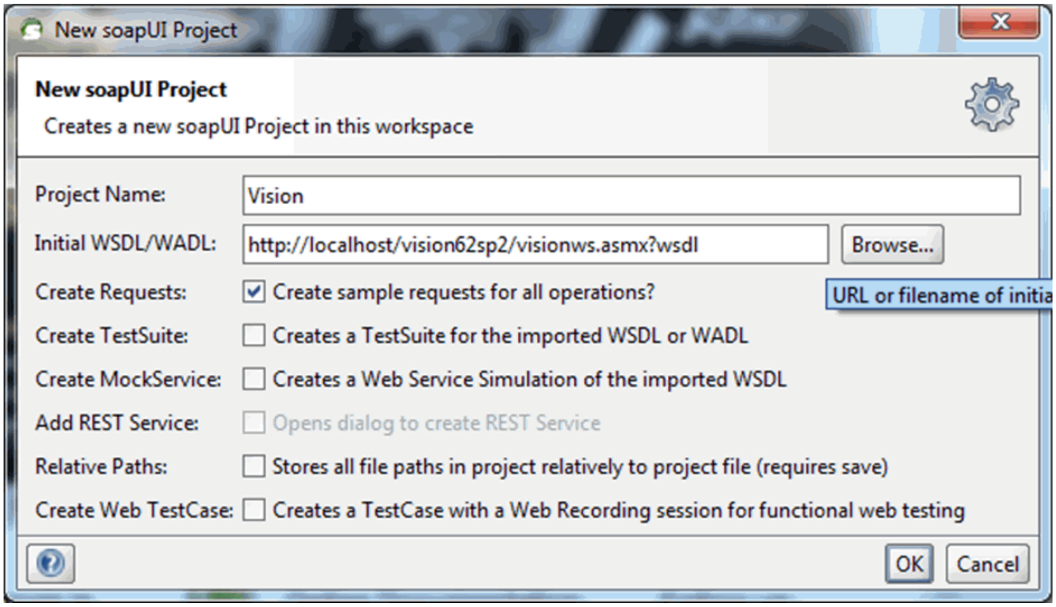


3. On the New soapUI Project dialog box, enter a project name, such as **Vision**.



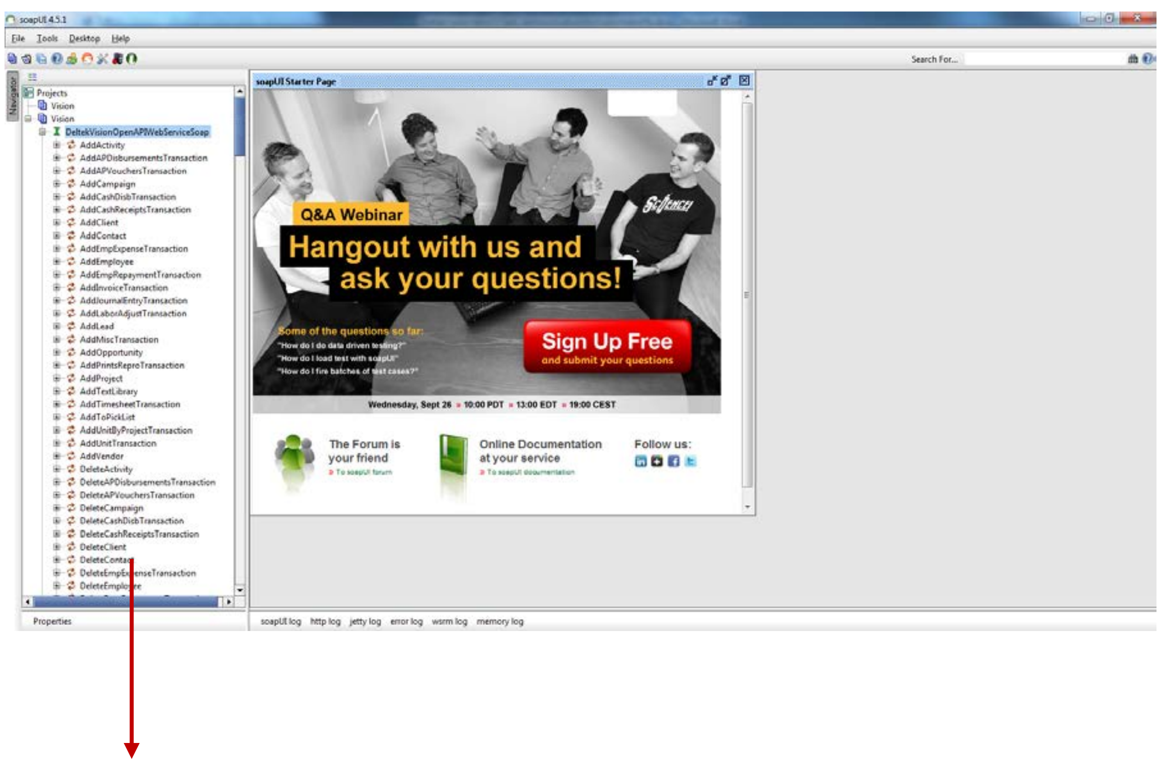
4. In the **Initial WSDL/WADL** field, enter the URL of your Vision web server and Vision directory plus the following: **/visionws.asmx?wsdl**.

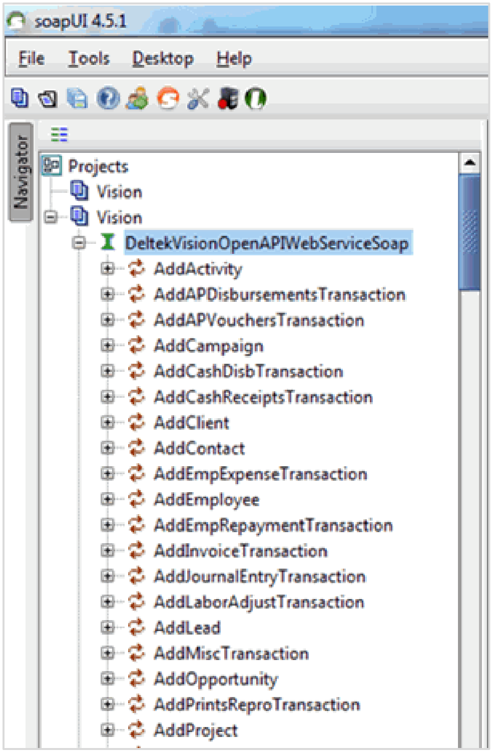
   Example: http://localhost/Vision70/visionws.asmx?wsdl



5. Select the **Create Requests: Create sample requests for all operations?** check box if it is not already selected.
6. Click **OK**.

   soapUI creates sample requests for every Vision API method in the Navigator pane on the left side of the next screen.





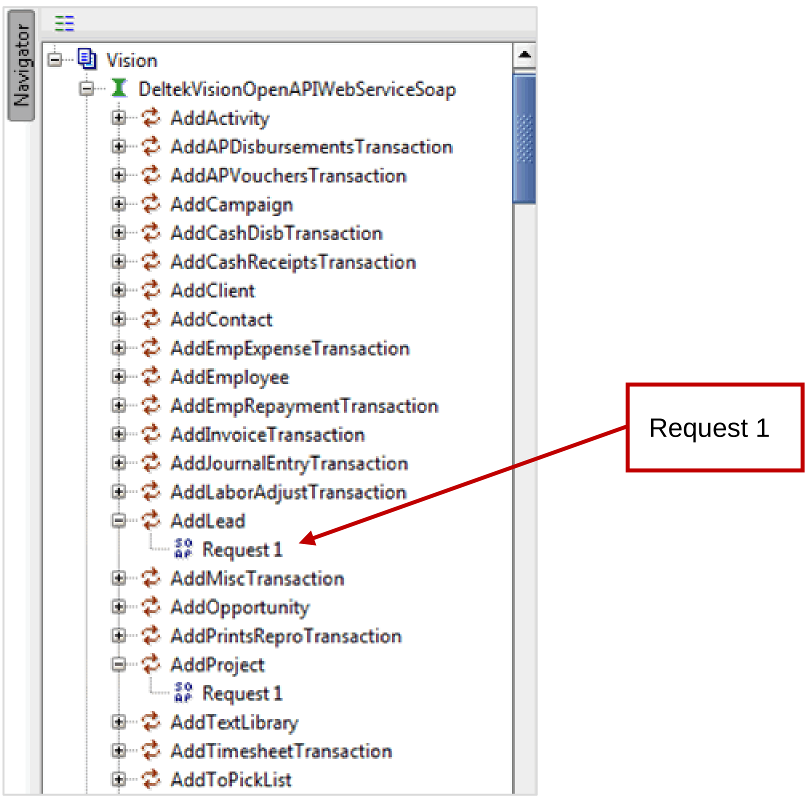
Now you can edit the sample requests to use for testing.

Edit a soapUI Sample Request to Test Vision APIs

When you use soapUI to call API methods, you take XML that soapUI has generated for a given call and add any parameters that you will be sending. You do this in the text editor in the center pane of the soapUI window. Because you selected the **Create sample requests for all operations?** check box when you created a new soapUI project, you can use a sample request as a starting point.

The Navigator pane in soapUI lists every method exposed by the API. If you expand a method in the Navigator pane, you see an existing sample request called **Request 1** directly below it.

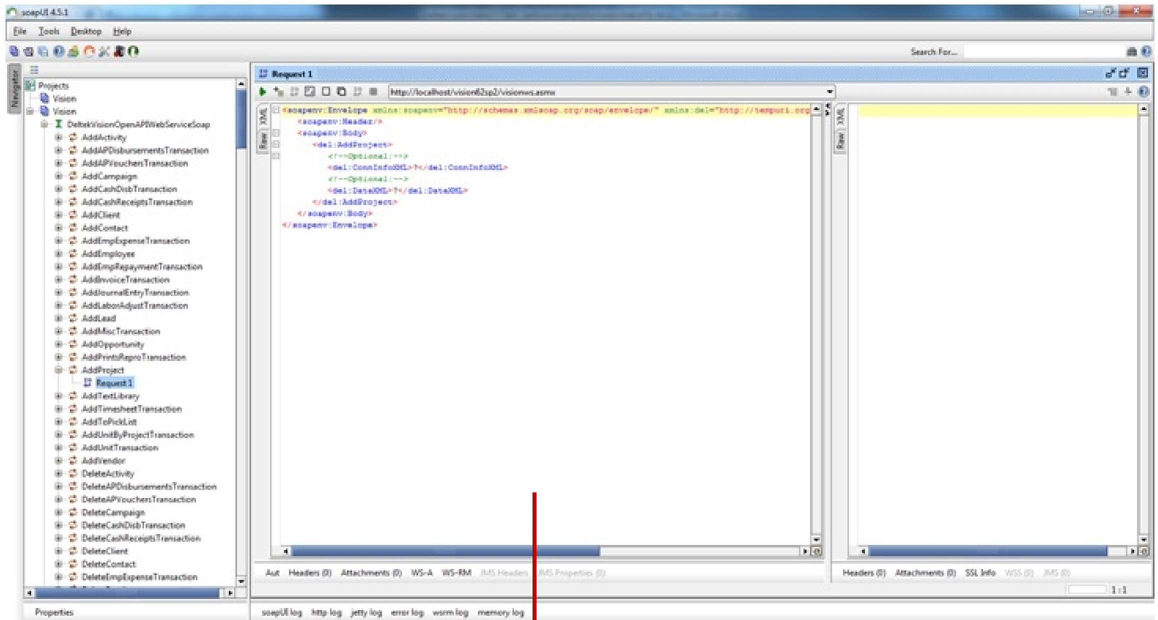
You can create as many sample requests for each method as you want in order to test different scenarios for the same call.

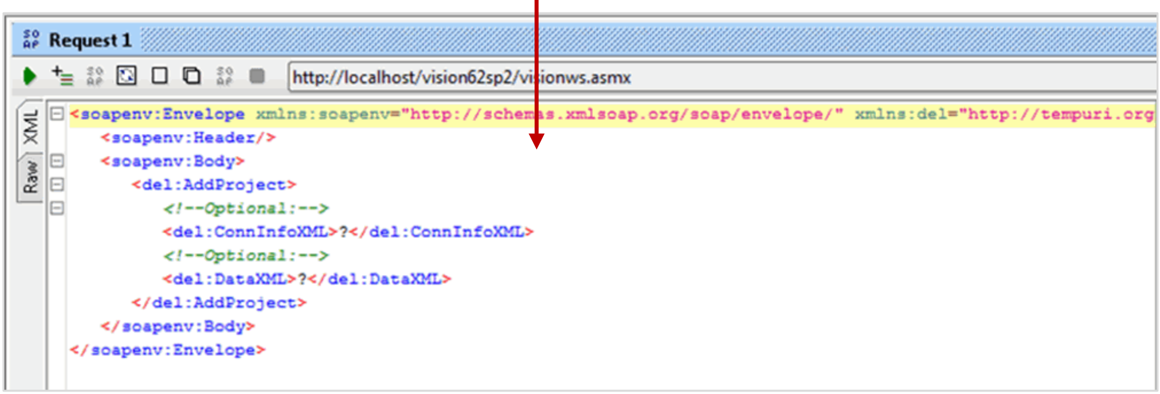


See the soapUI help for more information on how to create sample requests.

To edit a soapUI sample request, complete the following steps:

1. In the Navigator pane, click any method and expand it to display sample Request 1. The sample XML used by soapUI to call the method displays in the center text editor pane:





2. Edit the XML in the center pane to make a call for that method. In this example, we are making an AddProject call.
   - Replace the question with the proper parameters for that call.
   - Any parameters that pass in XML should wrap the XML in a CDATA block.
3. After you enter the information that you want to send to the API, click the green triangular play icon on the toolbar to execute the call. The results display in the pane on the right side of the screen.

Illustrated Examples

The following are examples of the XML to use in the soapUI text editor pane to test various Vision API methods.

Example of Connection XML Used for Testing APIs

You can modify the following XML to test Vision login validation and also to test Info Center access rights and record-level security:

```
<VisionConnInfo>
       <databaseDescription>VisionDemo30</databaseDescription>
       <userName>Admin</userName>
       <userPassword>test</userPassword>
       <integratedSecurity>N</integratedSecurity>
</VisionConnInfo>
```

Example of XML Data Used for APIs — Adding a New Lead Record

Use the XML below to test adding a new lead record.

There are two tables included in the XML because data is being added into only two tables, even though the actual schema has all of the associated/child tables included.

```
<?xml version="1.0"?>
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <REC>
              <Leads name="leads" alias="leads" keys="leadID"
codeTableColVal="code">
                     <ROW tranType="INSERT”>
                            <leadID>WSERV123</leadID>
                            <prefix>Mr.</prefix>
                            <firstName>Trevor</firstName>
                            <middleName>Kelly</middleName>
                            <lastName>Jackman</lastName>
                            <suffix>AIA</suffix>
                            <title>Chief Strategist</title>
                            <description>Test Lead Description</description>
                            <company>My Company, Inc.</company>
                            <email>jtrevor@myCompany.com</email>
                            <website>www.myComp.com</website>
                            <employee>000201</employee>
                            <source>Client Reference</source>
                            <status>sysNew</status>
                     </ROW>
              </Leads>
              <LeadCustomTabFields name="LeadCustomTabFields"
```

```
alias="LeadCustomTabFields" keys="LeadID">
             <ROW tranType="INSERT">
                    <LeadID>WSERV123</LeadID>
                    <custTest1Char>MyTest</custTest1Char>
                    <custTest2Num>23</custTest2Num>
                    <custTest3Char>Test</custTest3Char>
                    <custTest4Char></custTest4Char>
                    <custTest5Emp>000201</custTest5Emp>
                    <custTest6Lookup>Test1</custTest6Lookup>
             </ROW>
      </LeadCustomTabFields>
      </REC>
</RECS>
```

Even when a table is included in the XML, not all fields of the table need to be listed, with the exception of key fields. The only requirement is that fields be listed in the same order that they are in the schema. For example, Leads table XML can look like the following:

```
             <Leads name="leads" alias="leads" keys="leadID">
                    <ROW tranType="INSERT”>
                           <leadID>WSERV123</leadID>
                           <prefix>Mr.</prefix>
                           <firstName>Trevor</firstName>
                           <middleName>Kelly</middleName>
                           <lastName>Jackman</lastName>
                           <title>Chief Strategist</title>
                           <status>sysNew</status>
                    </ROW>
             </Leads>
```

To auto-generate leadID for a new record, use the **@Generate** option, as shown in the example below. The use of @Generate is applicable to other internally-generated keys as well (for example, **LinkID** in FileLinKs tables or **Seq** in grid type custom tab tables).

Using the Lead XML above for a Lead insert and the connection information in the “Example of Connection XML Used for Testing APIs” on page 10, you can see what this will look like in soapUI:

Notice the <![CDATA[]]> that wraps around each parameter. This is necessary only for parameters that contain XML.

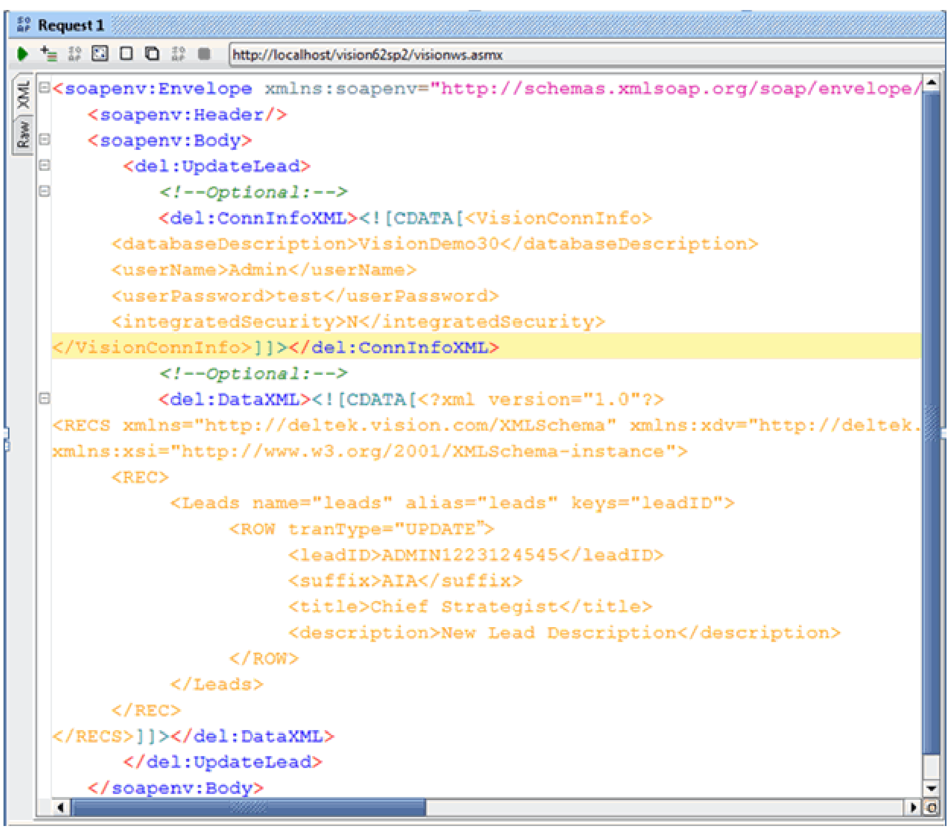
Example of XML Data Used for APIs — Updating an Existing Lead Record

The **tranType** attribute is set as **UPDATE.** Only fields that are being updated are included in XML.

In the example below, the fields that are being updated with a new value are **Suffix**, **Title**, and **Description**.

```
<?xml version="1.0"?>
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <REC>
                <Leads name="leads" alias="leads" keys="leadID">
                        <ROW tranType="UPDATE”>
                                <leadID>ADMIN1223124545</leadID>
                                <suffix>AIA</suffix>
                                <title>Chief Strategist</title>
                                <description>New Lead Description</description>
                        </ROW>
                </Leads>
        </REC>
</RECS>
```

The following is how it looks in soapUI:



Example of XML with Only Primary Key Values Used for Delete APIs

```
<?xml version="1.0"?>
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
	<REC>
		<Leads name="leads" alias="leads" keys="leadID">
			<ROW tranType="DELETE”>
				<leadID>WSERV123</leadID>
			</ROW>
		</Leads>
		<Leads name="leads" alias="leads" keys="leadID">
			<ROW tranType="DELETE”>
				<leadID>WSERV456</leadID>
			</ROW>
		</Leads>
	</REC>
</RECS>
```

Example of ValidationInfo.xml File

The following example is an excerpt from the ValidationInfo.xml file that includes information for the Leads table in the Lead Info Center. The file in its entirety includes information for all Vision Info Centers for all applicable tables. The content of this file is used to validate input XML data for Info Centers. The content of this file is configurable to some extent. You can include additional validation rules, such as requiring a field to be filled in, based on a specific type of role. This is similar to the feature that is available through the Info Center Designer.

```
<XML>
	<InfoCenter name="Leads">
		<Table name="Leads">
			<Column name="Status">
				<CodeTable>CFGLeadStatus</CodeTable>
				<CodeCol>Code</CodeCol>
				<DescCol>Description</DescCol>
			</Column>
			<Column name="Rating">
				<CodeTable>CFGLeadRating</CodeTable>
				<CodeCol>Code</CodeCol>
				<DescCol>Description</DescCol>
			</Column>
			<Column name="Industry">
				<CodeTable>CFGIndustry</CodeTable>
				<CodeCol>Code</CodeCol>
				<DescCol>Description</DescCol>
			</Column>
			<Column name="Employee">
```

```
                            <CodeTable>EM</CodeTable>
                            <CodeCol>Employee</CodeCol>
                            <DescCol>LastName</DescCol>
                    </Column>
            </Table>
    </InfoCenter>
</XML>
```

To set the **Employee** (field-title is ‘owner’ in the app) field that is required for the CRM type role, you would make the following change to the file:

```
                    <Column name="Employee" required=”CRM”>
                            <CodeTable>EM</CodeTable>
                            <CodeCol>Employee</CodeCol>
                            <DescCol>LastName</DescCol>
                    </Column>
```

To set a non-code table/foreign key-type field (‘Description’ column in Leads table) that is not listed in the file, as required for both CRM and ACCT type roles, you would add the following section to the file (for Leads table):

```
                    <Column name="Description" required=”ALL”>
                    </Column>
```

Note that because the **Description** field is not bound to a code table, the XML for this section is much simpler.

Example of Key Values Used with ‘SelectByKeys’ API

Clients Info Center (CL.ClientID is the primary key)

```
000000CAPE,0000MARBLE,00000DEVON
```

Employee Info Center (EM.Employee is the primary key)

```
000278,000002,000101
```

Projects Info Center (PR.WBS1, PR.WBS2 and PR.WBS3 are primary keys)

Example 1

Even though a single top-level project is being selected, key values for all three WBS columns are provided with WBS2 and WBS3 set to a single space. Notice that the composite key-value ends with a comma.

```
00000002000010.00; ; ,
```

Example 2

First key-value is a top level project, and the second is a phase. Again, the key values must end with a comma.

```
00000002000010.00; ; ,00000002000013.00;0003A; ,
```

Example 3

The key values for three, top-level projects are listed in the following example.

```
00000000000003.00; ; , 00000000000004.00; ; ,00000000000005.00; ; ,
```

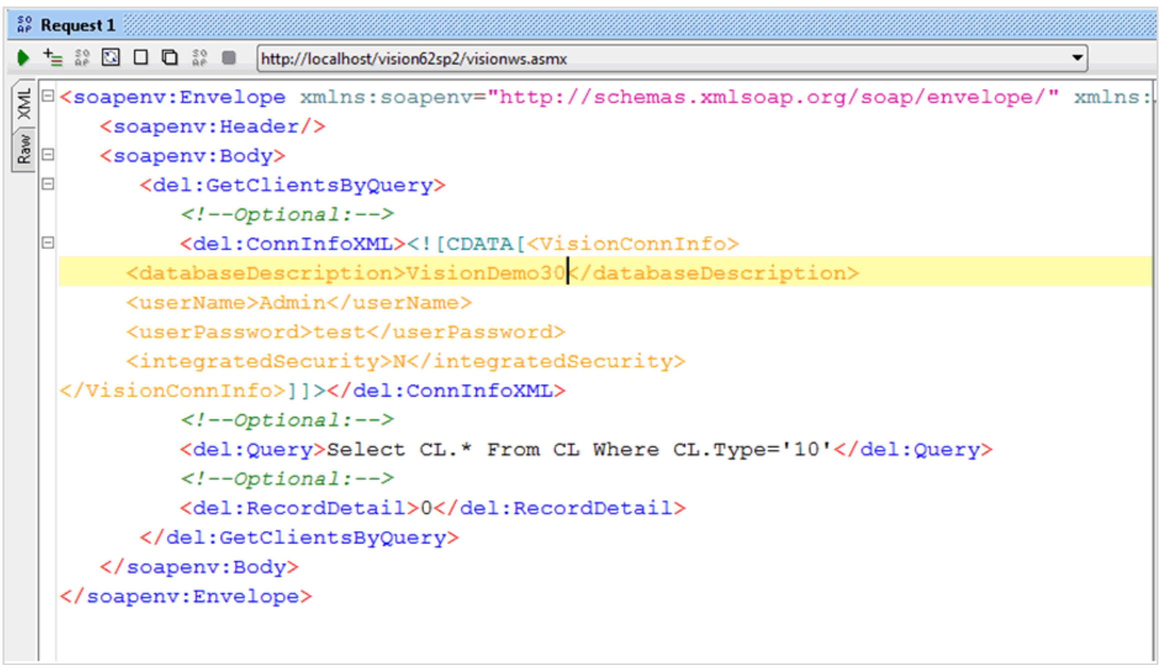
Example of Queries Used with ‘SelectByQuery’ API

Client Info Center

Example 1

```
Select CL.* From CL Where CL.Type=‘10’
```

The following is how example 1 would look in soapUI:



The following is the same example after running the GetClientsByQuery call. The results display in the pane on the right-hand side:

You can copy and paste the returned XML from the CDATA section (or save to a file), so you can view it can in an XML editor.

Example 2

```
Select CL.* From CL, CFGClientType Where CL.Type=CFGClientType.Code And
CFGClientType.Description='Health Care'
```

Project Info Center

Example 1

```
Select PR.* From PR Where PR.ProjectType='10'
```

Example 2

```
Select PR.* From PR, EMProjectAssoc Where PR.ChargeType='R' And

PR.WBS1=EMProjectAssoc.WBS1 And PR.WBS2=EMProjectAssoc.WBS2 And
PR.WBS3=EMProjectAssoc.WBS3 And EMProjectAssoc.Employee = '00004' And
EMProjectAssoc.RoleDescription like '%Structural Engineer%'
```

Using Data Returned by Select APIs to Test Add/Update/Delete APIs

When Select APIs are run, the returned XML data is always in a format that complies with the schema. However, it does not have any namespace references. In order to use the returned XML from Select APIs to test other APIs, namespace references must be added. The following is an example of what is returned by Select APIs:

```
<RECS>
       <REC>
               <CL name="CL" alias="CL" keys="ClientID">
                     <ROW>
                                   <ClientID>0000000PEA</ClientID>
                                   <Client>0000000PEA</Client>
                                   <Name>Peabody Industrial Group</Name>
                                   <Type>10</Type>
                                   <Status>A</Status>
                                   <ExportInd>Y</ExportInd>
                                   <WebSite>www.peabody.com</WebSite>
                                   <Memo></Memo>
```

```
                                  <CurrentStatus>Existing</CurrentStatus>
                                  <CustomCurrencyCode></CustomCurrencyCode>
                      </ROW>
               </CL>
       </REC>
</RECS>
```

To make it a valid XML for use with Add/Update/Delete APIs, the namespace references are added as shown in the following example. To add a new record, returned primary keys must be modified (for all tables in XML); otherwise new records cannot be added.

```
<?xml version="1.0"?>
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <REC>
               <CL name="CL" alias="CL" keys="ClientID">
                     <ROW tranType="INSERT">
                                   <ClientID>0000000PEA</ClientID>
                                  <Client>0000000PEA</Client>
                                  <Name>Peabody Industrial Group</Name>
                                  <Type>10</Type>
                                  <Status>A</Status>
                                  <ExportInd>Y</ExportInd>
                                  <WebSite>www.peabody.com</WebSite>
                                  <Memo></Memo>
                                  <CurrentStatus>Existing</CurrentStatus>
                                  <CustomCurrencyCode></CustomCurrencyCode>
                      </ROW>
               </CL>
       </REC>
</RECS>
```

Deltek is the leading global provider of enterprise software and information solutions for professional services firms, government contractors, and government agencies. For decades, we have delivered actionable insight that empowers our customers to unlock their business potential. Over 14,000 organizations and 1.8 million users in approximately 80 countries around the world rely on Deltek to research and identify opportunities, win new business, optimize resource, streamline operations, and deliver more profitable projects. Deltek – Know more. Do more.®

deltek.com