Deltek VisionXtend™

Test Client Application for Vision Web APIs / Web Services

Deltek Vision 6.1

Deltek
13880 Dulles Corner Lane
Herndon VA 20171
TEL: 703.734.8606
FAX: 703.734.1146

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical or technical errors may exist. Deltek, Inc. cannot accept responsibility for any kind of loss resulting from the use of this publication.

This page shows the original publication date. The information contained in this publication is subject to change without notice. Any improvements or changes to either the product or the book will be documented in subsequent updates.

This edition published June, 2009

Contents

Overview

Web APIs or Web services to access Info Centers in Vision are used by external applications that are created to make calls to the APIs. The development of these external applications requires effort and understanding of how to send data-to and also receive (and interpret) messages-from Vision Web APIs. In order to facilitate easy and convenient testing of Web APIs for Vision, a test client application with graphical user interface is included. This test application provides an easy way of interacting with Vision Web APIs and helps to understand how XML data is sent to Vision and what kind of XML messages are received in response.

The use of the test application is intended only for preliminary testing purposes and we do not recommend that it be used in an expanded capacity. Also, the test application has been created as a convenience, and does not cover the full scope of using or trying APIs in all scenarios.

Test Client Application Files

The following table lists the files that are located along with the test app executable. These files can be used with the test app after making proper changes to the contents of the files.

File Name	Description
VisionLogin.xml	File containing the connection/login XML that needs to be passed to Vision APIs. The content needs to be modified to match what is in the Vision database.
VisionURL.txt	A text file containing the URL of the Vision site, and it is automatically read by the app. The Vision URL is used by the client app to make calls to Web APIs.

For technical details regarding APIs, refer to the document *Web Services and API for Deltek Vision*.

The following files are sample XML data files that can be used with various Info Centers. The data included in these files is for testing purposes only and these files can be used as templates by API calling applications to send data to Vision through APIs.

File Name	Description
ClientTest.xml	A sample XML data file for Client Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
ContactTest.xml	A sample XML data file for Contact Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
ProjectTest.xml	A sample XML data file for Project Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
EmployeeTest.xml	A sample XML data file for Employee Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
OpportunityTest.xml	A sample XML data file for Opportunity Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
LeadTest.xml	A sample XML data file for Lead Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
MktCampaignTest.xml	A sample XML data file for Marketing Campaign Info Center and

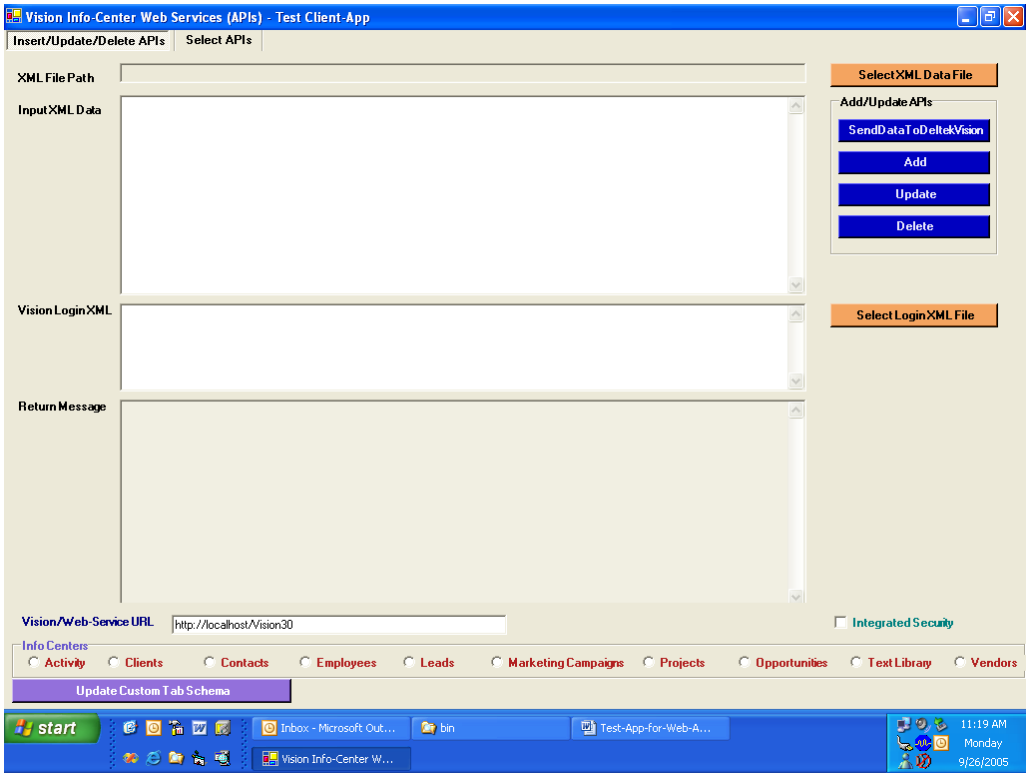
File Name	Description
	this includes all of the applicable association/child tables, except any user-defined fields/grids.
VendorTest.xml	A sample XML data file for Vendor Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
TextLibTest.xml	A sample XML data file for Text Library Info Center and this includes all of the applicable association/child tables, except any user-defined fields/grids.
ActivityTest.xml	A sample XML data file for activity area and this includes all of the applicable association/child tables.

Using the Test Client Application

For the convenience of using the test client application, the following are suggested:

- Set a location on the machine for the test client app (VisionAPITestClient.exe), and also for the XML files used for testing. This machine should not be your Vision Web server.
- Edit the VisionLogin.xml file so that it has the correct Vision username, password, and database description. The content of this file can also be modified during runtime within the test app. If your Vision Web server requires integrated authentication to log into Vision, ensure that you set the integrated security value to **Y** instead of **N**, and that your Windows login account name exists in Vision.
- If applicable, maintain a test results document to log results when specific API calls are made. The return messages will be in XML and the details of these messages are explained in the technical document *Web Services and API for Deltek Vision*.
- If the VisionURL.txt file is located in the same directory as that of the test app, the URL is automatically loaded when the application is run. The URL is displayed at the bottom.

Testing Add/Update/Delete APIs



Update Custom Tab Schema

Standard schema files for Info Centers in Vision do not include information regarding user-defined fields and grids, as these can be different in different databases. In order to include database-specific information regarding user-defined fields/grids into schemas, an API is available to update schema files located on the Vision Web server. The **Update Custom Tab Schema** button at the bottom left of the form calls this API for the Info Center selected, with the exception of Activity area. It is recommended that before using any Update/Add APIs, schemas for all Info Centers areas are updated with the current information on user-defined fields/grids.

To update Custom tab schema:

1. Select the Info Center (radio buttons at the bottom of the form; Activity is excluded).
2. Browse and open the VisionLogin.xml file.
3. Click **Update Custom Tab Schema**.

Add/Update APIs

Refer to the technical document *Web Services and API for Deltek Vision* regarding using the “SendDataToDeltekVision” API.

To add.update APIs:

1. Select the Info Center.
2. Click **Select XML Data File** to browse to and select the XML data file. The content of the file is loaded into the **Input XML Data** textbox. If the file VisionLogin.xml is located in the same directory, the connection XML from that file is automatically loaded into the **Vision Login XML** textbox. The XML data and connection XML can also be pasted directly into textboxes instead of browsing to the files.
3. Click **Select Login XML File** to browse to the VisionLogin.xml file. The connection XML is loaded into the **Vision Login XML** textbox. The connection XML can also be pasted directly into the textbox instead of browsing to the file.

Add New Records

Click **Add** to add records.

While adding new records, internally generated keys can be auto-generated by using the value as ‘@Generate’ (refer to the document *Web Services and API for Deltek Vision*). However, using ‘@Generate’ always generates a new key and it will add the same record over and over. On the other hand, a specific key value can be used to add the record as well. When no ‘@Generate’ is used, clicking on **SendDatatoDeltekVision** will also work. The ‘@Generate’ option is ideal for keys like ‘LinkID’ in file link tables or ‘Seq’ field in tables created for grid-type tabs.

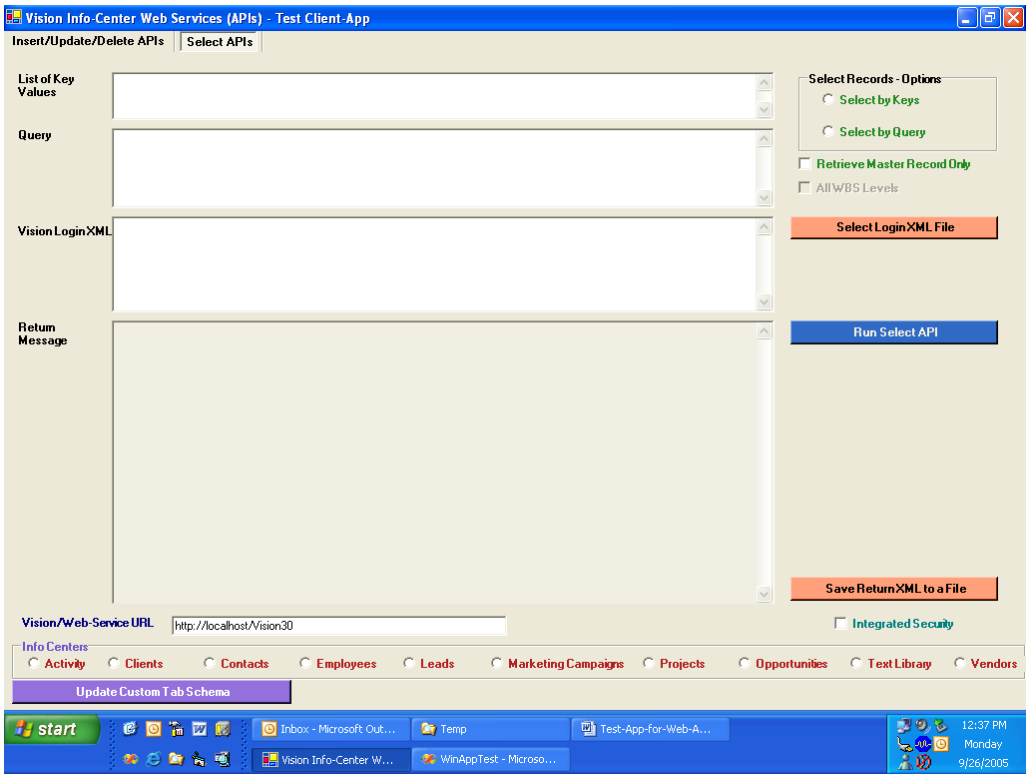
When ‘@Generate’ is used to generate new key values for Info Center records (like Client, Contact, Lead, etc.), the internally generated keys are returned once the records have been added to the database.

Update Existing Records

- In order to update existing records in Info Centers, the input XML should be such that it includes primary key fields and only those fields that need to be updated with a new value. The ‘tranType’ attribute needs to be set to ‘UPDATE’.
- **Caution**: Including empty field references in XML data will result in emptying of the existing data (if any) from the field in the database.
- Updating an existing record in an Info Center can also involve inserting/adding a new row into child or associated tables. This is still considered as an update to the main Info Center record. The ‘tranType’ attribute for such tables needs to be set to ‘INSERT’.
- The values for fields included in XML data for update are subject to validation before any updates are saved to the database.

Testing Select APIs

The parameters used for Select APIs are different from those used for APIs to add/update/delete records. The Select APIs need information as to the criteria that can be used to select specific records from the database. Like other APIs, Select APIs also require Info Center and connection (login and the database) information.



Select Record Options – Select by Keys and Select by Query

Basically there are two ways to specify the criterion used for retrieving data:

- Provide a list of individual key values that identify specific records.
- Give a query to retrieve a list of records based on the output of the query.

There are guidelines as to the required format for the keys or the query used. There are radio buttons in the test app to choose the select option of using keys or the query. Based on the select option chosen, a list of key values or a query is entered into the appropriate textbox.

- **Retrieve Master Record Only** — This option is used to specify whether the data from only the master table is returned, or the data from all associated/child tables are also returned. This option is not checked by default.
- **All WBS Levels —** There is another option that is applicable for retrieving project information only. The **All WBS Levels** checkbox is available only when Projects Info Center is selected and it is applicable for only the **Select by Query** option. When checked, information for all levels of the project (based on WBS1 returned from the query) is retrieved. It can be used in combination with the **Retrieve Master Record Only** option as well.

The Select API is run by clicking on **Run Select API**. When data in XML is returned, it is displayed in a textbox at the bottom, and the content can be saved into a file by clicking on **Save Return XML to a File**.

Integrated Security

If the Vision site is set up to use integrated windows security, the API calls need to be made along with client-side credentials. The **Integrated Security** checkbox on the form needs to be used under such circumstances. If integrated security option is being used to logon to Vision, the information in VisionLogin.xml needs to reflect it. The password can be empty if a Vision login is being used with integrated security.

Illustrated Examples

Example of Connection XML Used for Testing APIs

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

The above XML can be modified to test Vision login validation and also to test Info Center access rights and record-level security.

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

```
<?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>
```

Note that there are two tables included in the XML above. This is because of the fact that data is being added into only two tables even though the actual schema has all of the associated/child tables included.

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 something like this:

```
            <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>
```

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

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

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

```
<?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 'tranType' attribute is set as 'UPDATE'. Only fields that are being updated are included in XML. In the example above, the fields that are being updated with a new value are suffix, title, and description.

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

Below is an excerpt from the file and it is for Leads table in Leads Info Center. The file in its entirety does include information for all 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, and users 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 **Employee** (field-title is 'owner' in the app) field required for CRM type role, the following change would be made 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, the following section needs to be added 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, XML for the 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 is ended 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 are entered with a comma at the end.

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

Example 3

The key values for three top level projects are listed below.

```
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'
```

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, returned XML data is always in a format that complies with the schema, however it will not have any namespace references. In order to use the returned XML from Select APIs to test other APIs, namespace references need to 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>
```

In order to make it a valid XML for use with Add/Update/Delete APIs, namespace references are added as shown below.

In order to add a new record, returned primary key(s) need to 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>
```