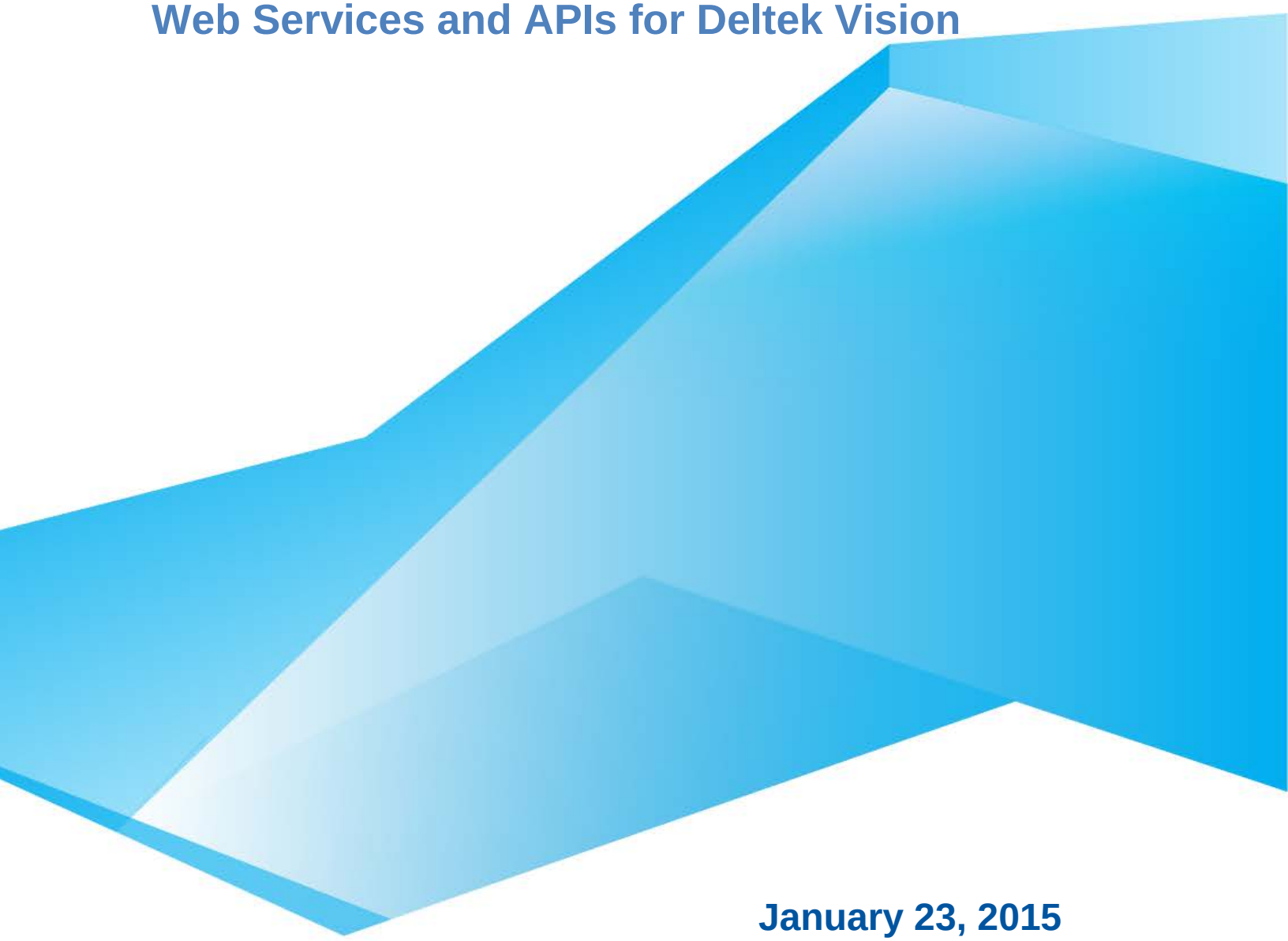
Deltek Know more. Do more.™

Deltek VisionXtend™ 7.4

Web Services and APIs for Deltek Vision



January 23, 2015

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This publication contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, or translated into another language, without the prior written consent of Deltek, Inc.

This edition published January 2015.

© 2015 Deltek, Inc.

Deltek's software is also protected by copyright law and constitutes valuable confidential and proprietary information of Deltek, Inc. and its licensors. The Deltek software, and all related documentation, is provided for use only in accordance with the terms of the license agreement. Unauthorized reproduction or distribution of the program or any portion thereof could result in severe civil or criminal penalties.

All trademarks are the property of their respective owners.

Contents

Overview 1
Adding Custom Notes to This Guide 1
If You Need Assistance 2
Web Services and APIs for Vision Info Centers 3
Web-Accessible APIs 3
List of Web Services/APIs 5
XML Schema for Vision Web Services/APIs 8
Application Development Notes 11
Return Messages 18
APIs to Retrieve Data from Vision Info Centers 22
APIs to Delete Info Center Records in Vision 31
Web Services and APIs for Accounting Transactions 35
XML Schema for Vision Transaction Web APIs 35
Schema Structure 35
Sample Schema 36
Sample Schema for the File Transaction Types that Are Affected by Multiple Tax Codes 38
Obtaining Schema Files for Transaction Areas 62
List of Vision Transaction Web APIs 63
Web APIs to Post Transactions 65
Transaction Security Validation 66
Transaction Data Validation 66
Multicompany Support for Transactions 67
Transaction API Return Messages 67
Integrating Web Services/API into an Application 71
Invoking a Web Service to Process Workflow Actions 73
Invoke Web Service Versus Invoke Custom Method Options in Workflow 73
Implementation Details 74
Application Development Notes 76
Invoking a Custom Method to Process Vision Workflow Actions 78
Implementation Details 78
Application Development Notes 80
Illustrated Examples with Code 82
Extending Data Validation Business Logic for Timesheets 85
Implementation Details 85

Timesheet Record in XML 85
Return Message 91
Vision Timesheet Data Validation Options 92
Extending Data Validation Business Logic for Expense Reports 93
Implementation Details 93
Expense Report Record in XML 93
Return Message 140
Vision Expense Report Data Validation Options 141
Enable Workflows for APIs 143
Connect for Microsoft API Calls 143
Add or Change the WebAPI.EnableWorkflow Setting 143

Overview

Vision incorporates service-oriented architecture (SOA) by supporting various Web services.

This documentation specifically applies to Vision Web services that provide Web-accessible application programming interfaces (APIs) that can be used by external applications to:

- Send data to and receive data from the Vision Info Centers.
- Process Vision accounting transactions.
- Submit timesheet records in Vision.
- Submit Vision expense reports.
- Trigger Vision workflows when Vision Info Center records are added or updated.

This document provides:

- Technical information about the available Web services and methods.
- Descriptions of how these Web services can be used by other applications.
- Examples of code to demonstrate proper usage of Web services.
- Descriptions of input and output messages handled by these Web services.
- 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 and expense reports.

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 adding 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 reading the document, cursor over a note icon to see the information. Double-click a note icon to edit the information.

If You Need Assistance

The Deltek Consulting Services Group provides assistance with custom development work that leverages the VisionXtend platform. Contact the services group at CustomServices@Deltek.com for more information about the services that they provide. Assistance with custom development work is not covered by your Vision Ongoing Support Plan (OSP).

Web Services and APIs for Vision Info Centers

The APIs exposed through Web services can carry out three types of database transactions related to Vision Info Centers:

- Insert
- Update
- Delete

Web services specific to each Info Center can be called separately to carry out any of the three types of transactions. A generic API is also available to run one or multiple types of transaction for any given Info Center by making a single call.

Web Services Description Language (WSDL) documentation for Web services provides critical information for developers to understand the interface and functional behaviors of APIs represented by Web services. WSDL Simple Object Access Protocol (SOAP) Binding is used to describe Web services. To generate WSDL documentation, use http://<*Vision server name*>/Vision/VisionWS.asmx?wsdl.

The proxy class file (VisionWS.asmx) for Web services is included with the Vision software. It is available for Vision Basic .NET (VB.NET) and C# languages. In addition, a compiled dynamic link library (DLL) is also available to access Web services.

Web-Accessible APIs

The following is a brief description of available Web-accessible APIs for Vision Info Centers.

SendDataToDeltekVision

This is a generic API that can be used to send data to any of Vision's supported Info Centers.

Each supported Info Center has two APIs available—one to add new records (insert) and the other to update existing records. For example, for the Employee Info Center, the APIs are AddEmployee() and UpdateEmployee().

The basic functional steps of all of the APIs are as follows:

1. **XML data validation**: Input XML data is validated against published schema.
2. **Vision login validation**: Checks whether or not Vision login credentials are valid.
3. **Vision record access validation**: Validates record-level access rights (add and modify) for the given Vision user.
4. **Data validation**: Input data is validated to ensure referential integrity and other database column level dependencies.
5. **Save data**.

The general format of the return message is shown below. The message format is in XML. The 'ReturnCode' is a number, and 'ReturnDesc' is a short description of the result returned by the service/API. The 'Detail' node consists of additional relevant information based on the return code.

```
<DLTKVisionMessage>
      <ReturnCode></ReturnCode>
      <ReturnDesc></ReturnDesc>
      <Detail></Detail>
</DLTKVisionMessage>
```

SendDataToDeltekVisionWithReturn

This method has the same functionality as the SendDataToDeltekVision method, plus it returns a snapshot of the changed records after the changes have been commited. It also supports the InfoCenterXML, KeysXML, and QueryXML parameters to support saving data to multiple tables and Info Centers with one method call.

The return message format is different from SendDataToDeltekVision. The format is shown below for changes sent to just one table. The RECS node is the same format as returned by the GetRecordsByKey call.

```
<DLTKVisionMessage>
      <ReturnCode></ReturnCode>
      <ReturnDesc></ReturnDesc>
      <REC></RECS>
</DLTKVisionMessage>
```

If changes are sent to multiple tables or Info Centers with one call, then the return format will look like the following:

```
<DLTKVisionMessage>
      <ReturnCode></ReturnCode>
      <ReturnDesc></ReturnDesc>
      <MULTIRECS>
            <REC></RECS>
            …
      </MULTIRECS>
</DLTKVisionMessage>
```

List of Web Services/APIs

All of the Web services use at least two or all of the following input parameters in Extensible Markup Language (XML) format:

- **InfoCenter** — This represents the generic Info Center name. For example, ‘Employees,’ ‘Projects,’ ‘Clients,’ and so on.
- **ConnInfoXML** — This is the Vision login information.
- **DataXML** — This is the data that needs to be saved to Vision database. It complies with the XML schema provided for the Info Center.

Return values for all services are sent in XML format as strings. Detailed information regarding parameters, messages-returned and their format can be found later in this document in the ”Application Development Notes" section. The following table contains a list of APIs available through Web services and their descriptions.

API and Description	
SendDataToDeltekVision(InfoCenter, ConnInfoXML, DataXML)	
Description:	This is a generic API that can add or update information in any Info Center. The value of the ‘InfoCenter’ parameter determines the Info Center area. DataXML must include values for internal key fields. DataXML can include data for multiple tables related to the Info Center. An example is adding an employee along with multiple degrees and skills for that employee.
SendDataToDeltekVisionWithReturn(InfoCenter, ConnInfoXML, DataXML)	
Description:	This method provides the same functionality as SendDataToDeltekVision but also returns snapshots of the records after the updates.
AddToPickList(PickListType, ConnInfoXML, DataXML)	
Description:	This adds items to code tables in the system. Not all code tables are supported.
AddActivity(ConnInfoXML, DataXML)	
Description:	This adds new activity records. DataXML does not need to include values for the internal key column ‘ActivityID.’ Internal keys are generated automatically, and they are passed back to the calling application after records are added.
UpdateActivity(ConnInfoXML, DataXML)	
Description:	This updates existing activity records. Multiple activity-related tables can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddCampaign(ConnInfoXML, DataXML)	
Description:	This adds new marketing-campaign records. DataXML does not need to include values for the internal key column ‘CampaignID.’ Internal keys are generated automatically, and they are passed back to the calling application after records are added.

API and Description	
UpdateCampaign(ConnInfoXML, DataXML)	
Description:	This updates existing marketing-campaign records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddClient(ConnInfoXML, DataXML)	
Description:	This adds new client records. DataXML does not need to include values for the internal key column 'ClientID.' Internal keys are generated automatically, and they are passed back to the calling application after records are added.
UpdateClient(ConnInfoXML, DataXML)	
Description:	This updates existing client records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddContact(ConnInfoXML, DataXML)	
Description:	This adds new contact records. DataXML does not need to include values for the internal key column 'ContactID.' Internal keys are generated automatically, and they are passed back to the calling application after records are added.
UpdateContact(ConnInfoXML, DataXML)	
Description:	This updates existing contact records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddEmployee(ConnInfoXML, DataXML)	
Description:	This adds new employee records. DataXML must include primary key values as specified in the XML schema.
UpdateEmployee(ConnInfoXML, DataXML)	
Description:	This updates existing employee records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddLead(ConnInfoXML, DataXML)	
Description:	This adds new lead records. DataXML does not need to include values for the internal key column 'LeadID.' Internal keys are generated automatically, and they are passed back to the calling application after records are added.

API and Description	
UpdateLead(ConnInfoXML, DataXML)	
Description:	This updates existing lead records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddOpportunity(ConnInfoXML, DataXML)	
Description:	This adds new opportunity records. DataXML does not need to include values for the internal key column 'OpportunityID.' Internal keys are generated automatically, and they are passed back to the calling application after records are added.
UpdateOpportunity(ConnInfoXML, DataXML)	
Description:	This updates existing opportunity records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddProject(ConnInfoXML, DataXML)	
Description:	This adds new projects. DataXML must include primary-key values as specified in the XML schema.
UpdateProject(ConnInfoXML, DataXML)	
Description:	This updates existing project records. The key values (internal or user-entered) for tables must be included in the XML input.
AddTextLibrary(ConnInfoXML, DataXML)	
Description:	This adds new text library records. DataXML must include primary-key values as specified in the XML schema.
UpdateTextLibrary(ConnInfoXML, DataXML)	
Description:	This updates existing text library records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.
AddVendor(ConnInfoXML, DataXML)	
Description:	This adds new vendor records. DataXML must include primary-key values as specified in the XML schema.
UpdateVendor(ConnInfoXML, DataXML)	
Description:	This updates existing vendor records. Multiple tables within a single Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input.

API and Description	
AddUDIC(ByVal ConnInfoXML As String, ByVal InfoCenter As String, ByVal DataXML As String)	
Description:	This adds new user-defined Info Center records. DataXML must include primary-key values as specified in the XML schema. "ByVal InfoCenter As String" is the name of the user-defined Info Center.
UpdateUDIC (ByVal ConnInfoXML As String, ByVal InfoCenter As String, ByVal DataXML As String)	
Description:	This updates existing user-defined Info Center records. Multiple tables within a single user-defined Info Center can be updated with a single call. The key values (internal or user-entered) for tables must be included in the XML input. "ByVal InfoCenter As String" is the name of the user-defined Info Center.

XML Schema for Vision Web Services/APIs

The data that you are adding or updating in the Vision database must be sent in XML format. The format of the XML data must comply with the schema. The order of the fields in your XML file must match the order of the fields that is defined by the schema.

If your XML file does not match the required schema and the order of the fields, you will receive an error when you use web services to update the Vision database.

Each applicable Info Center in Vision has an XML schema defined. Examples of the schema for each Info Center are included in schema files that are located on the Vision Web/app server in <*InstallDir*>\Vision\Web\Xsd directory (<InstallDir> is the directory where Deltek Vision is installed). The names of the schema files start with the generic Info-Center-name followed by '_Schema.xsd.' For example, the name of the XML schema file used for Employee Info Center would be 'Employee_Schema.Xsd.'

Handling Primary Key Columns that Use Internally Generated IDs

Some of the Info Centers have an internally generated unique key (GUID—globally unique identifier) as the primary key, and primary key values do not need to be provided when calling specific Web services to add new records. For example, AddClient, AddContact, and so on.

In order to have APIs generate unique values (GUIDs) for the internal primary key column on the fly, you must include a generic value '@Generate' for both the key attribute and key column. This generic value serves as the placeholder for the key value that is generated automatically.

The following Info Centers use an internal primary key, and the key columns in the database are listed below.

Info Center	Internal ID – Primary Key Column	Table
Client	ClientID	CL
Contact	ContactID	Contacts

Info Center	Internal ID – Primary Key Column	Table
Opportunity	OpportunityID	Opportunity
Lead	LeadID	Leads
Marketing Campaign	CampaignID	MktCampaign
Activity	ActivityID	Activity

The following is an example of XML input that specifies the use of auto-generated internal key values.

```
<RECS>
	<REC>
		<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>
					.
					.
			</ROW>
		<LeadCustomTabFields name="LeadCustomTabFields"
alias="LeadCustomTabFields" keys="LeadID">
			<ROW tranType="INSERT">
				<LeadID>@Generate</LeadID>
				<custTest1Char>MyTest</custTest1Char>
					.
					.
			</ROW>
		</LeadCustomTabFields>
		<LeadFileLinks name="LeadFileLinks" alias=" LeadFileLinks "
keys="LinkID,LeadID">
			<ROW tranType="INSERT">
				<LinkID>@Generate</LinkID>
				<LeadID>@Generate</LeadID>
				<Description>MyTestFile</Description>
					.
					.
			</ROW>
		</ LeadFileLinks >
	</REC>
</RECS>
```

The XML schema consists of constraints and rules to ensure that the incoming XML data is valid for Vision Web services to process. The schema needs to be carefully followed to make sure that the calling application is sending XML data in the right format.

The schema is designed to validate a number of things with respect to XML data that is sent as a parameter when calling Web services.

The following are some of the validations that are done:

- Data type and length/size
- Uniqueness of the key (within the XML ata)
- Mandatory columns/attributes
- Enumerated values

The basic format of the XML data in a simplistic representation looks like the following:

```
<RECS>
	<REC>
		<@TableName name="" alias="" keys="">
			<ROW transType="">
				<@key></@key>
				<@Column></@Column>
			</ROW>
		</@TableName>
		<@TableName name="" alias="" keys="">
			<ROW transType="">
				<@key></@key>
				<@Column></@Column>
			</ROW>
		</@TableName>
	</REC>
	<REC>
		<@TableName name="" alias="" keys="">
			<ROW transType="">
				<@key></@key>
				<@Column></@Column>
			</ROW>
		</@TableName>
	</REC>
</RECS>
```

- **@TableName** is the actual table name(s) depending on the Info Center for which the schema is defined.
- The **'@key'** and **'@Column'** references shown above are actual names of key columns and other columns respectively and they are different for different Info Centers. For example in the case of Client Info Center, @key will be replaced with 'clientID' and @Column will be replaced with actual column-names from the main client table (CL).

For a complete description of all the elements and the attributes, refer to schema documents containing the schema definitions.

Depending on the table, there can be one or more <ROW> elements. For the base or primary table, there can only be a single row. However, when certain tables share a one-to-many relationship with the primary table, there can be multiple <ROW> elements. The schema is designed such that it validates these restrictions and a validation message is returned when incoming XML data violates any of the constraints in the schema.

Internal keys (primary-key columns in Info Center base tables whose names end with 'ID') are not required if an Info-Center-specific Web service/API is used to Add records. Description of different API functions can be found later in this document. When internal key values are not provided, they are generated automatically by the Web service and new key values are returned after new records are added to the database.

Obtaining Latest Schema Files for Vision Info Centers

The latest schema files for Vision Info Centers can be obtained by using a Web service that returns schema file content (String).

Using API

You can use the following API to obtain schema file content for an Info Center:

GetSchema(ConnInfoXML, InfoCenter)	
Description:	Based on the value for 'InfoCenter' parameter for the call, content of the respective schema file is returned back.

The GenerateCustomTabSchema(ConnInfoXML, InfoCenter) API to refresh schema with custom tab information is not supported.

Application Development Notes

Input Parameters Used to Call Web Services

All parameters are passed in as strings.

InfoCenter

This indicates the name of the Info Center in Vision to and from which the data is being sent to and retrieved from.

While Info Center names are customizable within Vision application, the following generic names are used while calling Web services. Regardless of what the customized logical names are within Vision, the following Info Center names corresponding to those logical names within Vision must be used.

Info-Center Name*	Description	Primary Table	Key Column(s)
Projects	Project Info Center	PR	WBS1,WBS2,WBS3
Clients	Client Info Center	CL	ClientID
Contacts	Contact Info Center	Contacts	ContactID
Employees	Employee Info Center	EM	Employee
Opportunities	Opportunity Info Center	Opportunity	OpportunityID
Leads	Leads Info Center	Leads	LeadID

Info-Center Name*	Description	Primary Table	Key Column(s)
MktCampaigns	Mkt. Campaign Info Center	MktCampaign	CampaignID
Vendors	Vendor Info Center	VE	Vendor
TextLibraries	TextLibrary Info Center	TextLibrary	Name
Activities	Activity Info Center	Activity	ActvityID

***** These are used for the 'InfoCenter' parameter while calling Web services.

The key-column names that end with 'ID' use internally generated key values.

ConnInfoXML

This is Vision login information that is used to authenticate the calling application's access to Vision database. The following is the XML format of this parameter:

```
<VisionConnInfo>
      <databaseDescription>VisionDemo30</databaseDescription>
      <userName>Admin</userName>
      <userPassword>test</userPassword>
<encPassword>
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855</encPassword>
      <integratedSecurity>N</integratedSecurity>
<SessionID>d1116ddbd324407e89524f9cb4477831</SessionID>
</VisionConnInfo>
```

Parameter components:

- **Database description (databaseDescription)** — This is the description of the Vision database that has been entered into 'weblink' utility for Vision. If multiple Vision databases are in use, this particular parameter identifies the Vision database that needs to be used for Web-service transactions.
- **User name (username)** — This is the name of the Vision user from Vision application security. In addition to using this Vision login to authenticate calling application's access to Vision database, the access rights of the Vision security role that the user belongs to, is used to determine whether or not new records can be added, and as to which records can be modified.
- **User password (userPassword)** — This is the password stored in Vision database.
- **Encrypted password (encPassword)** — This is the encrypted password hash stored in the Vision database. It is based on SHA-256 hash.

  The username (in upper case) and the password are combined for the password hash.

Example:

Vision username: Password

Vision password: Password1

The SHA-256 hash is based on the following string: PASSWORDPassword1

Hash value:
42edadc1235401632a2b63792d37b7f89cf61851d3ac15c72b731316604b499a

If the password is blank, then the username is not used, and the hash value is:
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

- **Integrated security** — This indicates whether or not the integrated Windows login from the calling machine needs to be used to validate Vision login. Allowable values are **Y** and **N**.
- **Session ID** — This is the value of the session ID passed back from a previous call. Passing this back allows the API to reuse sessions for faster and more efficient calls.

DataXML

This is input data in XML format that is in compliance with the supplied schema.

The following is a sample DataXML string that can be passed to 'SendDataToDeltekVision' API to add a new lead record. This XML string includes internal-primary-key value (for leadID). This example consists of a single row/record, but data XML can contain any number of rows/records.

```
<?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="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>
                        <statusReason/>
                        <industry>01</industry>
                        <rating>01</rating>
                        <recordStatus>A</recordStatus>
                        <address1>1234 Test Drive</address1>
                        <address2>Suite 123</address2>
                        <address3/>
                        <city>Pleasantville</city>
                        <state>MD</state>
                        <zip>22343</zip>
                        <country>USA</country>
                        <businessPhone>703-989-8878</businessPhone>
                        <businessFax>703-989-8879</businessFax>
                        <mobile>703-989-8899</mobile>
                        <home/>
                        <pager>703-989-8877</pager>
                  </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>
```

Project Info Center Example

You must include the yellow-highlighted text, otherwise the save operation will fail. No value is required.

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <REC>
    <PR name="PR" alias="PR" keys="WBS1,WBS2,WBS3"
xmlns="http://deltek.vision.com/XMLSchema">
      <ROW tranType="INSERT">
        <WBS1>R2013000.034</WBS1>
        <WBS2> </WBS2>
        <WBS3> </WBS3>
        <Name>20130904 API AutoNumber-Project</Name>
        <ChargeType>R</ChargeType>
        <SubLevel>N</SubLevel>
        <Org>0A:0A:BO:AR</Org>
        <OpportunityID></OpportunityID>
       </ROW>
    </PR>
  </REC>
</RECS>
```

Opportunity Info Center Example

You must include the yellow-highlighted text, otherwise the save operation will fail. No value is required.

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <REC>
    <Opportunity name="Opportunity" alias="Opportunity"
keys="OpportunityID" xmlns="http://deltek.vision.com/XMLSchema">
      <ROW tranType="INSERT">
        <OpportunityID>@Generate</OpportunityID>
        <Opportunity>OPPAPI0000000012</Opportunity>
        <Name>API Opportunity</Name>
        <PRWBS1></PRWBS1>
      </ROW>
    </Opportunity>
```

```
  </REC>
</RECS>
```

Example of Updating Records in an Info Center

The following example shows multiple Project Info Center records and a user-defined field being updated. In addition, a row from the Employee grid on the Team tab in the Project Info Center is being deleted.

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <REC>
        <PR name="PR" alias="PR" keys="WBS1,WBS2,WBS3">
           <ROW tranType="UPDATE">
            <WBS1>00000000.016</WBS1>
            <WBS2>AAA</WBS2>
            <WBS3> </WBS3>
            <SubLevel>N</SubLevel>
                 <ReimbAllow>789</ReimbAllow>
            <BillingCurrencyCode>USD</BillingCurrencyCode>
                 </ROW>
        </PR>
        <EMProjectAssoc name="EMProjectAssoc" alias="EMProjectAssoc"
keys="RecordID">
               <ROW tranType="DELETE">
                <RecordID>F709458DB8FA4B0489143D9F3A059C84</RecordID>
              </ROW>
         </EMProjectAssoc>
  <ProjectCustomTabFields name="ProjectCustomTabFields"
alias="ProjectCustomTabFields" keys="WBS1,WBS2,WBS3">
           <ROW tranType="UPDATE">
            <WBS1>00000000.016</WBS1>
            <WBS2>AAA</WBS2>
            <WBS3> </WBS3>
            <CustManager2>0C0001</CustManager2>
               </ROW>
  </ProjectCustomTabFields>
 </REC>
<REC>
```

```
      <PR name="PR" alias="PR" keys="WBS1,WBS2,WBS3">
         <ROW tranType="UPDATE">
          <WBS1>00000000.012</WBS1>
          <WBS2>A000</WBS2>
          <WBS3>T000</WBS3>
          <SubLevel>N</SubLevel>
               <ReimbAllow>555</ReimbAllow>
          <BillingCurrencyCode>USD</BillingCurrencyCode>
                </ROW>
      </PR>
      <EMProjectAssoc name="EMProjectAssoc" alias="EMProjectAssoc"
keys="RecordID">
             <ROW tranType="DELETE">
              <RecordID>F709458DB8FA4B0489143D9F3A059C84</RecordID>
             </ROW>
      </EMProjectAssoc>
   <ProjectCustomTabFields name="ProjectCustomTabFields"
alias="ProjectCustomTabFields" keys="WBS1,WBS2,WBS3">
             <ROW tranType="UPDATE">
              <WBS1>00000000.012</WBS1>
              <WBS2>A000</WBS2>
              <WBS3>T000</WBS3>
              <CustManager2>0A0101</CustManager2>
                 </ROW>
   </ProjectCustomTabFields>
  </REC>
</RECS>
```

Modifying the Address for a Client Record

If you modify the address for a client in the Vision Client Info Center using a Web Services API, you may receive the error message: **Cannot update Address description in for the Client IC records**.

The following is the correct way to update the client **Address** fields using Web Services APIs:

1. Assign the original value that is entered in the **Address** field to the "origVal" attribute.
2. Assign the new value for the **Address** field to the "new_val" attribute.

Important: You must assign the original value of the **Address** field to the element.

Example:

```
<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="UPDATE">
        <ClientID>ADVCLIENTA000000CVS</ClientID>
        <Client>0000000CVS</Client>
        <Name>API Updated Consumer Value Stores-CVS</Name>
      </ROW>
    </CL>
    <CLAddress name="CLAddress" alias="CLAddress"
keys="ClientID,Address">
      <ROW tranType="UPDATE">
        <ClientID>ADVCLIENTA000000CVS</ClientID>
        <Address origVal="MainContact"
new_val="NewMainContact">MainContact</Address>
        <Address1>API Updated Address1</Address1>
        <Address2>API Updated Address2</Address2>
      </ROW>
</CLAddress>
  </REC>
</RECS>
```

Return Messages

As specified previously, return values or messages are sent back in a standard XML format. The calling application can use this information to verify whether the API/Service call was successful or whether it had any problems. The structure of the return message makes it easy for the application to parse and process it and determine the next action based on what was returned from Vision. The format is as follows:

```
<DLTKVisionMessage>
      <ReturnCode></ReturnCode>
      <ReturnDesc></ReturnDesc>
      <Detail></Detail>
</DLTKVisionMessage>
```

Components are:

- **Return-Code** is a short-code, and it is predefined to indicate success or the type of error.
- **Return-Desc** is a brief description of the message or the event that happened during the call.
- The **Detail node** consists of either further detailed description of the error or return values that correspond to transaction.
- **Return-Code** is always “1” when the service has been successfully run with no errors.

Vision Record Access Validation

In Vision, access to Info Centers in terms of adding new records and updating existing records is controlled at the ‘role’ level. Record level security is assigned to a role, and it controls the list of records that a user belonging to that role can edit or update. All Web-accessible API calls validate record access rights in Vision with respect to the login/user name specified in connection-XML (ConnXML) parameter.

The following is the format of the returned error message when record access validation fails. The <AddRec> section indicates whether or not there is a restriction to add new records. The <UpdateRec> section provides information on failed record level security validation. The <ReturnValues> section under <UpadateRec> lists key values of all of the records that are not available to update.

```
<DLTKVisionMessage>
	<ReturnCode>ErrRecAccess</ReturnCode>
	<ReturnDesc>Record Access Validation Error</ReturnDesc>
	<Detail>
		<AddRec>
			<Message>Permission to add new records is denied</Message>
		</AddRec>
		<UpdateRec>
			<Message>Permission to update records is denied</Message>
			<ReturnValues>
				<keyVal>ADMIN1074277037011</keyVal>
				<keyVal>ADMIN1086620567592</keyVal>
			</ReturnValues>
		</UpdateRec>
	</Detail>
</DLTKVisionMessage>
```

Validation of XML Data

Incoming XML data is validated to ensure that all of the business rules that are either application-enforced or user-defined are completely complied with. For each of the records in each of the tables, the values for applicable columns are checked against the business rules. If any of the column values are not in compliance with the business rules, a return message in a specified format is passed back to the calling application. The validation is completed for the entire input XML data before returning the validation result. The validation result would therefore consist of information about all values for various columns that do not satisfy the business rules.

The following is the general format of the validation result that is returned. There can be multiple <Message> nodes in the validation result depending on how many fields and how many rows had validation problems.

```
<DLTKVisionMessage>
	<ReturnCode>ErrDataVal</ReturnCode>
	<ReturnDesc>Data Validation Error</ReturnDesc>
	<Detail>
		<ValidationResult>
			<Message>
				<Table></Table>
				<Column></Column>
				<ColValue></ColValue>
				<RowNum></RowNum>
				<PrimaryRecKeyVal></PrimaryRecKeyVal>
				<MsgCode></MsgCode>
				<MsgDesc></MsgDesc>
			</Message>
		</ValidationResult>
	</Detail>
</DLTKVisionMessage>
```

Setting Fields as 'Required'

Fields can be set as required based on the role type (CRM, Accounting, or Administrator) by making changes to ValidationInfo.xml file found in <VisionInstallDir>\Web\Xml directory. The settings in this file are used only for Vision APIs. The file also has information used to ensure implicit and explicit referential integrity constraints. For example, a file contains information to ensure that a field that is bound to a code table accepts only those values that are already present in the code table.

The following is the format of the ValidationInfo.xml file:

```
<XML>
	<InfoCenter name="@InfoCenter">
		<Table name="@TableName">
			<Column name="@ColumnName" required="@RoleType">
				<CodeTable>@ReferenceTable</CodeTable>
				<CodeCol>@ReferenceColumn</CodeCol>
				<DescCol>@DescriptionColumn</DescCol>
			</Column>
			<Column name="@ColumnName" required="@RoleType">
			</Column>
		</Table>
	</InfoCenter>
</XML>
```

Components are:

- **@InfoCenter** — This is the Info Center name.
- **@TableName** — This is the name of the table related to the Info Center, which is also referenced in the schema.
- **@ColumnName** — This is the name of the column in the table indicated by @TableName.

- **@RoleType** — This is the role type from Vision security. The possible values are 'CRM,' 'ACCT,' and 'Admin.' When the 'required' attribute is set as 'ALL,' the field will be required for both CRM and ACCT type roles.
- **@ReferenceTable** — This is the name of the master or reference table against which the data for the column in question needs to be verified.
- **@ReferenceColumn** — This is the name of the column in @ReferenceTable.
- **@DescriptionColumn** — The name of the column containing the description for code/key in **@ReferenceColumn**. This is not currently being used and it is optional.

As shown above, the nodes <CodeTable>, <CodeCol>, and <DescCol> are optional if a column needs to be set as required but does not have any referential integrity to validate.

The following is an example of what an excerpt of the file ValidationInfo.xml may look like:

```
<XML>
	<InfoCenter name="Projects">
		<Table name="PR">
			<Column name="Org" required="ACCT">
				<CodeTable>Organization</CodeTable>
				<CodeCol>Org</CodeCol>
				<DescCol>Name</DescCol>
			</Column>
			<Column name="ClientID" required="CRM">
				<CodeTable>CL</CodeTable>
				<CodeCol>ClientID</CodeCol>
				<DescCol>Name</DescCol>
			</Column>
			<Column name="ProjectType">
				<CodeTable>CFGProjectType</CodeTable>
				<CodeCol>Code</CodeCol>
				<DescCol>Description</DescCol>
			</Column>
			<Column name="Description" required="CRM">
			</Column>
		</Table>
	</InfoCenter>
</XML>
```

The ValidationInfo.xml file can be edited to set 'required' attribute for fields that are either already in the file or for new fields that have been added. Changes must be carefully made, and it is strongly recommended that the entries for fields that are a part of the standard ValidationInfo.xml file must not be removed or altered.

Custom Tab Fields Validation

In addition to validating application-defined validation rules, any user-defined validation rules on custom-tab fields are also validated. Following are the user-defined validation constraints that are validated.

- The 'Required' constraint defined based on Vision role-type.
- For numeric values, minimum and maximum bounds are checked to ensure that the value is in the correct range.
- Lookup type fields with the option 'LimitToList' set.
- For fields that have a default-value specified, the default value is automatically uploaded into the database if no value is provided in the input XML data.

For each column value that violates validation rules, a <Message> node is added under <ValidationResult>. Each message node consists of table name, column name, row number (as appears in the XML data), the primary key value for the main Info Center record, message code (MsgCode), and message description (MsgDesc) information.

The message code is a predefined code that represents the kind of validation error that has occurred. The message description is a brief text message that explains the error.

The following table provides a list of message codes and descriptions for validation messages that are included in the <Detail> section:

Validation error code (<MsgCode>)	Validation error description (<MsgDesc>)
VAL1001	Value not in the master/code table
VAL1002	Value is required
VAL1003	Min value constraint not met
VAL1004	Max value constraint not met

The design of the XML schema allows input XML data to omit non-primary tables and fields when adding or updating records. For validation purposes, only those fields that are included in the data-XML are validated, and corresponding validation messages are returned. Deltek recommends that if all required field constraints need to be validated, all of the fields in the table be included in XML data, whether or not they have data.

APIs to Retrieve Data from Vision Info Centers

There are two basic APIs to get record level information from Info Centers in Vision. There are also Info Center-specific APIs to retrieve data from respective Info Centers. In terms of functionality, both of the basic APIs and the Info Center-specific APIs do exactly the same thing. The difference is with respect to calling parameters. Generic APIs require InfoCenterXML as one of the parameters.

The following APIs use parameters that have their own XML structure: InfoCenterXML, KeysXML and QueryXML. The following section explains the structure of these parameters.

Parameter	Structure
InfoCenterXML	This parameter can either be the name of an Info Center or XML with the following structure. Required nodes and attributes are in dark red. <InfoCenters> <InfoCenter ID=”1” Name=”Clients” Table=”CL” RowAccess=”1” PartialAccess = “1” Chunk=”1” ChunkSize=”100”> <CL> <ClientID/> <Name/> … <RecordCount/> </CL> </InfoCenter> … </InfoCenters> **Nodes:** ▪ **InfoCenters** — This required node encloses one or more InfoCenter nodes. ▪ **InfoCenter** — This required node represents a request for data from a specific Info Center. The InfoCenter node has the following attributes: ▪ **ID** — This is required. It can be any string and is used to uniquely identify the return result when there are more than one InfoCenter nodes. ▪ **Name** — The is required. It is the name of the Info Center to request data from. ▪ **Table** — The is optional. It is the name of a specific table within the Info Center from which to request data. If this attribute exists, then only results from this table will be returned. ▪ **RowAccess** — This is optional. This is a Boolean (1\0) attribute that will return the Vision access right for each record. The access rights are returned by an extra field at the end of each row called RA. This field will have a value of either R for read-only, U for update-only, and A for full access. ▪ **PartialAccess** — This is optional. This is a Boolean (1\0) attribute that will return records even if they fall outside of a user's Vision read-security settings. The returned records will contain only the primary key and one descriptive field.

Parameter	Structure
	▪ **Chunk** — This is optional. This is a numeric field for requesting data in “chunks.” It represents which chunk of data is being requested. ▪ **ChunkSize** — This is optional. This is a numeric attribute that sets the number of records to be returned with the current chunk. The number of records returned may be less than this if it is the last chunk. This attribute is required if the Chunk attribute is used. ▪ **TableName** — This is an optional node. If the Table attribute is used in the InfoCenter node, then under the InfoCenter node there can be a node with the name of the table. This is used to specify the fields to be returned. Normally, all fields specified in the schema for a given Info Center and table are always returned. This node can be used to return just certain fields. Under this node should be a node with the name of each field requested. Optionally, there can be just a single RecordCount node which will then return a record count from the table.
KeysXML	This parameter can either be a single key value, a comma delimited list of key values, or XML with the following structure. Required nodes and attributes are in dark red. <KeyValues> <Keys ID=”1”> <Key> <Fld>KEYVALUE</Fld> … </Key> … </Keys> … </KeyValues> **Nodes:** ▪ **KeyValues** — This required node encloses one or more Keys nodes. ▪ **Keys** — This required node encloses a list of keys. You must have one of these for each InfoCenter node in the corresponding InfoCenterXML. The Keys node has the following attribute: ▪ **ID** — This is required. The value of this attribute must be the same as the ID attribute for the corresponding InfoCenter node. ▪ **Key** — This required node encloses a single key. ▪ **Fld** — This required node encloses the value of the key. For multi-part keys there will be a Fld node for each part.

Parameter	Structure
QueryXML	This parameter can either be query text or XML with the following structure. All node and attributes are required. <Queries> <Query ID=”1>select * from CL</Query> </Queries> ▪ **Queries** — This encloses one or more Query nodes. ▪ **Query** — This encloses the query text for the corresponding InfoCenter node. Each Query node must have the following attribute: ▪ **ID** — The value of this attribute must be the same as the ID attribute for the corresponding InfoCenter node.

Generic APIs

The following generic APIs are described below:

- GetRecordsByKey
- GetRecordsByQuery
- GetUDICByKey
- GetUDICByQuery

Generic APIs and their Descriptions	
GetRecordsByKey(ConnInfoXML, InfoCenterXML, KeysXML, RecordDetail)	
Description:	The set of records retrieved is based on the key value/values specified in ‘keys’ parameter. Records are returned in XML format complying with the schema for the particular Info Center. The ‘RecordDetail’ parameter determines if only the record from the main table is retrieved or if information from all child/association tables are also returned.
Parameters:	▪ **ConnInfoXML** — Connection information to validate Vision login and the database access. ▪ **InfoCenter** — Info Center name as specified in the beginning of this document. ▪ **Keys** — Primary key values separated by a comma. It can also be a single key value to retrieve a single record. ▪ **RecordDetail** — If left empty, records from all of the association and child tables are retrieved. To retrieve only the basic information (record from the main Info Center table), use a value of ‘Primary’ with this parameter.
GetRecordsByQuery(ConnInfoXML, InfoCenterXML, QueryXML, RecordDetail)	

Description	The set of records retrieved is based on the key value/values specified in 'keys' parameter. Records are returned in XML format complying with the schema for the particular Info Center. The 'RecordDetail' parameter determines if only the record from the main table is retrieved or if information from all child/association tables are also returned.
Parameters	▪ **ConnInfoXML** — Connection information to validate Vision login and the database access. ▪ **InfoCenter** — Info Center name as specified in the beginning of this document. ▪ **Query** — A SQL statement selecting records from the main Info Center table. The query can have nested and correlated queries to base selection on any of the related tables. The required format of the query would be: SELECT <Info Center main table>.* FROM <Info Center main table> WHERE <Where Clause> FROM clause can include other tables in addition to the main table depending on how complex the Where clause is. Example query for Employee Info Center: SELECT EM.* FROM EM, Organization WHERE EM.Org = Organization.Org AND **Organization.Name** = 'My Company' ▪ **RecordDetail** — If left empty, record(s) from all of the association and child tables are retrieved. To retrieve only the basic information (record from the main Info Center table only), a value of 'Primary' is used with this parameter. **Note** In the case of Projects Info Center, the record detail parameter can have one of the following four values: ▪ **Empty ("")** — Data from all child/associated tables are retrieved but for only the top level project. ▪ **Primary** — Data from only the main Info Center table (PR) is retrieved for only the top level project. ▪ **AllPrimary** — Data from only the main Info Center table (PR) is retrieved for all WBS levels of the project. **All** — Data from all child/associated tables is retrieved for all WBS levels of the project.
Description	The set of records retrieved is based on the key value/values specified in 'keys' parameter. Records are returned in XML format complying with the schema for the particular user-defined Info Center. The 'RecordDetail' parameter determines whether only the record from the main table is retrieved or whether information from all child/association tables are also returned.
Parameters	▪ **ConnInfoXML** — Connection information to validate Vision login and the

	database access. ▪ **InfoCenter** — User-defined Info Center name. ▪ **Keys** — Primary key values separated by a comma. It can also be a single key value to retrieve a single record. ▪ **RecordDetail** — If left empty, records from all of the association and child tables are retrieved. To retrieve only the basic information (record from the main user-defined Info Center table only), use a value of 'Primary' with this parameter.
GetUDICByQuery(ConnInfoXML, InfoCenter, Query, RecordDetail)	
Description	The set of records retrieved is based on the key value/values specified in 'keys' parameter. Records are returned in XML format complying with the schema for the particular user-defined Info Center. The 'RecordDetail' parameter determines whether only the record from the main table is retrieved or whether information from all child/association tables are also returned.
Parameters	▪ **ConnInfoXML** — Connection information to validate Vision login and the database access. ▪ **InfoCenter** — The user-defined Info Center name. ▪ **Query** — A SQL statement selecting records from the main user-defined Info Center table. The query can have nested and correlated queries to base selection on any of the related tables. The required format of the query would be: SELECT <user-defined Info Center main table>.* FROM <user-defined Info Center main table> WHERE <Where Clause> FROM clause can include other tables in addition to the main table depending on how complex the WHERE clause is. Example query for "Recruits" user-defined Info Center: SELECT Recruits.* FROM Recruits, Organization WHERE Recruits.Org = Organization.Org AND Organization.Name = 'My Company' ▪ **RecordDetail** — If left empty, records from all of the association and child tables are retrieved. To retrieve only the basic information (record from the main user-defined Info Center table only), a value of 'Primary' is used with this parameter.

Info Center-Specific APIs to Retrieve Data

All records are returned in XML format, complying with the schema for the particular Info Center. The 'RecordDetail' parameter determines whether only the record from the main table is retrieved or whether information from all child/association tables are also returned.

For more details regarding parameters used for these APIs, refer to the section above where generic APIs to retrieve data from Vision are described.

GetProjectsByKey(ConnInfoXML, Keys, RecordDetail)

Description:	A set of records is retrieved based on the key (PR.WBS1, PR.WBS2, PR.WBS3) value or values specified in 'keys' parameter.
GetProjectsByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetClientsByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (CL.ClientID) value or values specified in 'keys' parameter.
GetClientsByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetContactsByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (Contacts.ContactID) value or values specified in 'keys' parameter.
GetContactsByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetEmployeesByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (EM.Employee) value or values specified in 'keys' parameter.
GetEmployeesByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetOpportunitiesByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (Opportunity.OpportunityID) value or values specified in 'keys' parameter.
GetOpportunitiesByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetVendorsByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (VE.Vendor) value or values specified in 'keys' parameter.
GetVendorsByQuery(ConnInfoXML, Query, RecordDetail)	

Description:	A set of records is retrieved based on the query.
GetLeadsByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (Leads.LeadID) value or values specified in 'keys' parameter.
GetLeadsByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetMktCampaignsByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (MktCampaign.CampaignID) value or values specified in 'keys' parameter.
GetMktCampaignsByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetTextLibrariesByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (TextLibrary.Name) value or values specified in 'keys' parameter.
GetTextLibrariesByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetActivitiesByKey(ConnInfoXML, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (Activity.ActivityID) value or values specified in 'keys' parameter.
GetActivitiesByQuery(ConnInfoXML, Query, RecordDetail)	
Description:	A set of records is retrieved based on the query.
GetUDICByKey(ConnInfoXML, InfoCenter, Keys, RecordDetail)	
Description:	A set of records is retrieved based on the key (<*user-defined Info Center name*>,<*user-defined Info Center name*>ID) value or values specified in 'keys' parameter.
GetUDICByQuery(ConnInfoXML, InfoCenter, Query, RecordDetail	
Description:	A set of user-defined Info Centers is retrieved based on the query.

Other APIs to Retrieve Data

GetPicklist(ConnInfoXML, PicklistInfoXML)

GetPicklist(ConnInfoXML, PicklistInfoXML)	
Description:	This retrieves one or more picklists from Vision. Picklists are returned.
Parameters:	**ConnInfoXML** – Standard connection info parameter. **PicklistInfoXML** – XML that defines what picklists to return. The following is an example of the format with the required nodes in dark red: <PickListRequest> <PickList Type=”CFGActivityType” /> … </PickListRequest> **PickList node** – There is one PickList node for each requested picklist. **Type attribute** – This required attribute corresponds to a specific picklist used in Vision. The following picklist types are supported: CFGActivitySubject CFGActivityType CFGChargeType CFGClientCurrentStatus CFGClientRole CFGClientStatus CFGClientType CFGContactRole CFGContactTitle CFGContactType CFGCountry CFGEmployeeRelationship CFGEmployeeRole CFGEmployeeStatus CFGOpportunitySource CFGOpportunityStage CFGOpportunityStatus CFGOpportunityType CFGPhoneFormat CFGPrimarySpecialty CFGProbability CFGProjectStatus CFGStates CFGVendorRole CFGVendorStatus CFGVendorType ContactStatus Organization Payterms SEUser
Returns:	One or more picklist results corresponding to each requested type in the

GetPicklist(ConnInfoXML, PicklistInfoXML)	
	PicklistInfoXML. The return XML has the following structure: <RECS> <REC> <PickListType1_Picklist Type="PickListType1" internalId="fld1" description="fld2"> <ROW><fld1>ID1</fld1><fld2>Desc1</fld2></ROW> … </PickListType1> </REC> … </RECS> There will be one REC and one PickListType_Picklist node for each type requested. The PickListType_Picklist node will have the "PickListType" part replaced with the actual picklist type (for example, CFGActivityType_Picklist). The internalId and description attributes define which fields are used for the code and description parts of the picklist, respectively. There may also be other fields returned, depending on the picklist.

APIs to Delete Info Center Records in Vision

For each of the Info Centers there is an API available to delete records. All delete APIs accept two parameters, similar to APIs used to add and update records. The two parameters used with delete APIs are listed below.

- **ConnInfoXML** — This is Vision login information that is used to authenticate the calling application's access to the Vision database.
- **DataXML** — This is input data in XML format. This is in compliance with the supplied schema for the Info Center.

Unlike APIs used for adding and updating Info Center records, the DataXML for delete APIs are required to include only data for key fields for the main table of the Info Center. It is important that the 'tranType' (transaction type) attribute in the XML data is set to 'DELETE'. Also, data for only the main table of the Info Center is required.

If the DataXML string includes the data for non-key fields and non-primary tables of the Info Center, it will be stripped off automatically before processing the data for deletion of records.

Return Message

If deletion is successful, a return value of '1' (as explained in a previous section of this document) is returned. In the event of problems, appropriate error messages in the standard return-message XML format are returned.

List of Delete APIs

DeleteRecords(ConnInfoXML, DataXML)	
Description:	This deletes records from any number of tables and/or Info Centers. DataXML is required to include XML data for the table and the key fields listed below for each specific Info Center delete method.
DeleteClient(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **CL** Key Field: **ClientID**
DeleteContact(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **Contacts** Key Field: **ContactID**
DeleteEmployee(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **EM** Key Field: **Employee**
DeleteProject(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **PR** Key Fields: **WBS1, WBS2, WBS3** **Notes**

	▪ As specified in the schema, the 'keys' attribute needs to include values for all three key fields, separated by a semicolon. ▪ Individual keys are separated by a comma. ▪ Because Projects Info Center uses more than one key field and a default value of space is used for wbs2 and wbs3 fields when the corresponding level for the project does not exist, the list of keys should always end with a comma and spaces need to be included when applicable. For example, 0100000.00;02; ,0200000.00; ; ,
DeleteOpportunity(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **Opportunity** Key Field: **OpportunityID**
DeleteLead(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **Leads** Key Field: **LeadID**
DeleteCampaign(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **MktCampaign** Key Field: **CampaignID**
DeleteVendor(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the

	key fields listed below: Primary Table: **VE** Key Field: **Vendor**
DeleteTextLibrary(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **TextLibrary** Key Field: **Name**
DeleteActivity(ConnInfoXML, DataXML)	
Description:	Records whose primary key information is included in DataXML are deleted from the Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: **Activity** Key Field: **ActivityID**
DeleteUDIC(ByVal ConnInfoXML As String, ByVal InfoCenter As String, ByVal DataXML As String	
Description:	Records whose primary key information is included in DataXML are deleted from the user-defined Info Center. DataXML is required to include XML data for the table and the key fields listed below: Primary Table: ***<user-defined Info Center>*** Key Field: ***<user-defined Info Center>*ID** “ByVal InfoCenter As String” is the name of the user-defined Info Center.

Web Services and APIs for Accounting Transactions

A complete set of Web APIs is available to add, edit, delete, and post all types of accounting transactions in Vision. The supported transactions for APIs, which you enter in Vision Transaction Entry, are:

- A/P Disbursements
- A/P Vouchers
- Cash Disbursements
- Cash Receipts
- Employee Expenses
- Employee Repayments
- Invoices
- Journal Entries
- Labor Adjustments
- Miscellaneous Expenses
- Prints and Reproductions
- Timesheets
- Units
- Units by Project

For each of the transaction types, there are APIs to add transaction entries, edit existing entries, delete all or part of unposted transaction entries, and post transactions.

XML Schema for Vision Transaction Web APIs

The transaction data that needs to be added, updated, deleted, or posted in the Vision database is sent in XML format. The format of the XML data must comply with the schema provided.

Each applicable transaction type in Vision has an XML schema defined. You will find these schema files on the Vision Web/app server in <InstallDir>\Vision\Web\Xsd directory (<InstallDir> is the directory where Deltek Vision is installed). The names of the schema files start with the name of the transaction type followed by '_Transaction_Schema.xsd.' For example, the name of the XML schema file used for A/P Vouchers type transactions is 'APVouchers_Transaction_Schema.Xsd.'

Schema Structure

The schema for each transaction type maps to three entities in the database, based on the way transaction information is stored in the Vision database. These three entities share the common names Control, Master, and Detail. The entities are prefixed with letters that indicate the type of transaction. For example, the names of the entities for A/P Voucher type transactions are apControl, apMaster, and apDetail.

The Control entity tracks the primary details of the transaction file. The Master entity keeps track of different components of the transaction file such as invoices and checks. The Detail entity stores all of the details or line-items of the transaction.

In terms of the relationship between these entities, each transaction file is represented by a unique entry (transaction file), and the Control entity can have one or more entries or rows in the Master entity. Each record (for example, invoice and check) in the Master entity can have one or more rows in the Detail entity.

For more information on fields and columns included in entities that are used for various types of transactions in Vision, see the Vision data dictionary in Vision online help. On the title bar the top of the Vision main screen, click Help, and then click **Data Dictionary**.

Sample Schema

The following is an example of a schema structure. This particular example shows the schema structure for an A/P Voucher transaction in Vision. Each transaction's (file's) information is included within a <REC> element. The XML data that is sent to Vision can have data for one or more transactions (files), each included in a separate <REC> element.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2006 sp2 U (http://www.altova.com)-->
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://deltek.vision.com/XMLSchema
C:\DeltekVision\Vision41\Web\xsd\APVouchers_Transaction_Schema.xsd">
       <REC>
              <apControl keys="Batch" name="apControl" alias="apControl">
                     <ROW tranType="">
                            <Batch/>
                            <Recurring/>
                            <Posted/>
                            <Creator/>
                            <Period/>
                            <EndDate/>
                            <Total/>
                            <DefaultLiab/>
                            <DefaultBank/>
                            <DefaultDate/>
                            <PostPeriod/>
                            <PostSeq/>
                            <Company/>
                            <DefaultCurrencyCode/>
                     </ROW>
              </apControl>
              <apMaster keys="Batch,MasterPKey" name="apMaster"
alias="apMaster">
                     <ROW tranType="">
                            <Batch/>
```

```
                        <MasterPKey/>
                        <Vendor/>
                        <InvoiceDate/>
                        <Invoice/>
                        <TransDate/>
                        <LiabCode/>
                        <BankCode/>
                        <PayTerms/>
                        <PayDate/>
                        <Address/>
                        <Posted/>
                        <Seq/>
                        <Voucher/>
                        <CurrencyCode/>
                        <CurrencyExchangeOverrideMethod/>
                        <CurrencyExchangeOverrideDate/>
                        <CurrencyExchangeOverrideRate/>
                        <BarCode/>
                        <PaymentExchangeOverrideMethod/>
                        <PaymentExchangeOverrideDate/>
                        <PaymentExchangeOverrideRate/>
                </ROW>
        </apMaster>
        <apDetail keys="Batch,MasterPKey,PKey" name="apDetail"
alias="apDetail">
                <ROW tranType="">
                        <Batch/>
                        <MasterPKey/>
                        <PKey/>
                        <Seq/>
                        <Description/>
                        <WBS1/>
                        <WBS2/>
                        <WBS3/>
                        <Account/>
                        <Amount/>
                        <SuppressBill/>
                        <NetAmount/>
                        <CurrencyExchangeOverrideRate/>
                        <PONumber/>
                        <PaymentExchangeRate/>
                        <PaymentAmount/>
                        <PaymentExchangeInfo/>
                </ROW>
        </apDetail>
    </REC>
</RECS>
```

Sample Schema for the File Transaction Types that Are Affected by Multiple Tax Codes

The following transaction types are affected by multiple tax codes:

- A/P Vouchers
- A/P Disbursements
- Employee Expenses
- Cash Disbursements
- Journal Entries

The following key columns are different, depending on the transaction type:

Table	Table Description	Columns
xxControlDefaultTaxCodes (optional)	This is the list of default tax codes for the transaction batch.	Columns are: ▪ Batch — The transaction batch filename. ▪ TaxCode — The ID of the tax code. ▪ Seq — The numeric sequence of the tax code.
xxDetailTax (optional)	This is the list of tax codes that are associated with each of the transaction detail lines. The entry is linked to xxDetail via the “keys” list in the schema. For example, apDetailTax is linked to apDetail via the “Batch,MasterPKey,PKey” combination.	Columns are: ▪ Batch — The batch from the xxDetail row. ▪ MasterPKey — The MasterPKey from the xxDetail row. ▪ PKey — The PKey from the xxDetail row. ▪ TaxCode — The tax code that is associated with the detail line. ▪ TaxAmount — The tax amount. ▪ CompoundOnTaxCode — The **Tax Code to Include in Basis** value in Tax Codes Configuration. ▪ Seq — The numeric sequence of the list of tax codes that are associated with the detail line.

In the following examples, the tax data is the yellow-highlighted text.

A/P Voucher Transactions

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <REC>
  <apControl name="apControl" alias="apControl" keys="Batch">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <Recurring>Y</Recurring>
    <Posted>N</Posted>
    <Creator>MTC</Creator>
    <Period>201308</Period>
    <EndDate>2013-08-26T00:00:00.000</EndDate>
    <Total>0</Total>
    <DefaultLiab/>
    <DefaultBank/>
    <DefaultDate>2013-08-26T00:00:00.000</DefaultDate>
    <PostPeriod>0</PostPeriod>
    <PostSeq>0</PostSeq>
    <Company>0A</Company>
    <DefaultCurrencyCode>USD</DefaultCurrencyCode>
    <SubmittedBy/>
    <SubmittedDate/>
   </ROW>
  </apControl>

  <apControlDefaultTaxCodes name="apControlDefaultTaxCodes"
alias="apControlDefaultTaxCodes" keys="Batch,TaxCode">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <TaxCode>Y15</TaxCode>
    <Seq>1</Seq>
   </ROW>
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
```

```
      <TaxCode>AGS</TaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>PERC10</TaxCode>
      <Seq>3</Seq>
    </ROW>
  </apControlDefaultTaxCodes>
<apMaster name="apMaster" alias="apMaster" keys="Batch,MasterPKey">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <Vendor>AMASSACH.SU</Vendor>
      <InvoiceDate>2013-08-25T00:00:00.000</InvoiceDate>
      <Invoice>12001</Invoice>
      <TransDate>2013-08-26T00:00:00.000</TransDate>
      <LiabCode>Atrade</LiabCode>
      <BankCode>USD4USD</BankCode>
      <PayTerms>Date</PayTerms>
      <PayDate>2013-09-24T00:00:00.000</PayDate>
      <Address>&lt;Default&gt;</Address>
      <Posted>N</Posted>
      <Seq>2</Seq>
      <Voucher/>
      <CurrencyCode>BHD</CurrencyCode>
      <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
      <CurrencyExchangeOverrideDate/>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <BarCode/>
      <PaymentExchangeOverrideMethod>N</PaymentExchangeOverrideMethod>
      <PaymentExchangeOverrideDate/>
      <PaymentExchangeOverrideRate>0</PaymentExchangeOverrideRate>
      <PaymentCurrencyCode>BHD</PaymentCurrencyCode>
      <Status>N</Status>
      <AuthorizedBy/>
```

```
      <RejectReason/>
      <ModUser/>
      <ModDate/>
    </ROW>
  </apMaster>
  <apDetail name="apDetail" alias="apDetail" keys="Batch,MasterPKey,PKey">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <Seq>1</Seq>
      <Description/>
      <WBS1>2006AAR0003</WBS1>
      <WBS2>USTK</WBS2>
      <WBS3/>
      <Account>51200</Account>
      <Amount>560.00</Amount>
      <SuppressBill>N</SuppressBill>
      <NetAmount>500.0</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <PONumber/>
      <PaymentExchangeRate>0</PaymentExchangeRate>
      <PaymentAmount>560.00</PaymentAmount>
      <PaymentExchangeInfo/>
      <ExpenseCode/>
      <OriginatingVendor/>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <Seq>2</Seq>
      <Description/>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
```

```
      <Account>52311</Account>
      <Amount>1300.12</Amount>
      <SuppressBill>N</SuppressBill>
      <NetAmount>1000.0</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <PONumber/>
      <PaymentExchangeRate>0</PaymentExchangeRate>
      <PaymentAmount>1300.12</PaymentAmount>
      <PaymentExchangeInfo/>
      <ExpenseCode/>
      <OriginatingVendor/>
    </ROW>
  </apDetail>

  <apDetailTax name="apDetailTax" alias="apDetailTax"
keys="Batch,MasterPKey,PKey,TaxCode">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>LOW3HIGH12</TaxCode>
      <TaxAmount>15</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>15</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
```

```
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>AGS</TaxCode>
      <TaxAmount>30</TaxAmount>
      <CompoundOnTaxCode>AHI</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>25</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>LOW4HIGH11</TaxCode>
      <TaxAmount>100</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y11</TaxCode>
      <TaxAmount>100.12</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
  </apDetailTax>
 </REC>
```

```
</RECS>
```

A/P Disbursement Transactions

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <REC>
  <cvControl name="cvControl" alias="cvControl" keys="Batch">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <Recurring>Y</Recurring>
    <Selected>N</Selected>
    <Posted>N</Posted>
    <Creator>MTC</Creator>
    <Period>201308</Period>
    <EndDate>2013-08-26T00:00:00.000</EndDate>
    <Total>0</Total>
    <DefaultLiab/>
    <DefaultBank/>
    <DefaultDate>2013-08-28T00:00:00.000</DefaultDate>
    <PostPeriod>0</PostPeriod>
    <PostSeq>0</PostSeq>
    <Company>0A</Company>
    <DefaultCurrencyCode>USD</DefaultCurrencyCode>
    <SubmittedBy/>
    <SubmittedDate/>
   </ROW>
  </cvControl>

  <cvControlDefaultTaxCodes name="cvControlDefaultTaxCodes"
alias="cvControlDefaultTaxCodes" keys="Batch,TaxCode">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <TaxCode>Y15</TaxCode>
    <Seq>1</Seq>
   </ROW>
   <ROW tranType="INSERT">
```

```
      <Batch>MTC-API</Batch>
      <TaxCode>AGS</TaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>PERC10</TaxCode>
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>PERC6</TaxCode>
      <Seq>4</Seq>
    </ROW>
  </cvControlDefaultTaxCodes>

 <cvMaster name="cvMaster" alias="cvMaster" keys="Batch,CheckNo">
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <CheckNo>000001234001</CheckNo>
    <TransDate>2013-08-28T00:00:00.000</TransDate>
    <Vendor>AMASSACH.SU</Vendor>
    <InvoiceDate>2013-08-18T00:00:00.000</InvoiceDate>
    <Invoice>88888</Invoice>
    <BankCode>ASF Env.</BankCode>
    <Address>&lt;Default&gt;</Address>
    <Posted>N</Posted>
    <LiabCode/>
    <Seq>1</Seq>
    <Voucher/>
    <BarCode/>
    <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
    <CurrencyExchangeOverrideDate/>
    <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
    <CurrencyCode>USD</CurrencyCode>
    <Status>N</Status>
```

```
      <AuthorizedBy/>
      <RejectReason/>
      <ModUser/>
      <ModDate/>
    </ROW>
  </cvMaster>
  <cvDetail name="cvDetail" alias="cvDetail" keys="Batch,CheckNo,PKey">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <Seq>1</Seq>
      <Description>Line 1</Description>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
      <Account>52101</Account>
      <Amount>1100</Amount>
      <SuppressBill>N</SuppressBill>
      <NetAmount>1000</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <PaymentAmount>1100</PaymentAmount>
      <ExpenseCode/>
      <OriginatingVendor/>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <Seq>2</Seq>
      <Description>Line 2</Description>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
      <Account>52100</Account>
      <Amount>3300</Amount>
```

```
      <SuppressBill>N</SuppressBill>
      <NetAmount>3000</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <PaymentAmount>3200</PaymentAmount>
      <ExpenseCode/>
      <OriginatingVendor/>
    </ROW>
  </cvDetail>

  <cvDetailTax name="cvDetailTax" alias="cvDetailTax"
keys="Batch,CheckNo,PKey,TaxCode">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>LOW3HIGH12</TaxCode>
      <TaxAmount>40</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
     <PKey>MTCAPIPKey1</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>35</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>AGS</TaxCode>
      <TaxAmount>25</TaxAmount>
      <CompoundOnTaxCode>AHI</CompoundOnTaxCode>
```

```
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>100</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>LOW4HIGH11</TaxCode>
      <TaxAmount>120</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000001234001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y11</TaxCode>
      <TaxAmount>5</TaxAmount>
      <CompoundOnTaxCode>NY_TAX</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
  </cvDetailTax>
 </REC>
</RECS>
```

Cash Disbursements

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<REC>
 <cdControl name="cdControl" alias="cdControl" keys="Batch">
  <ROW tranType="INSERT">
   <Batch>MTC-API</Batch>
   <Recurring>Y</Recurring>
   <Posted>N</Posted>
   <Creator>MTC</Creator>
   <Period>201308</Period>
   <EndDate>2013-08-26T00:00:00.000</EndDate>
   <Total>0</Total>
   <DefaultBank/>
   <PostPeriod>0</PostPeriod>
   <PostSeq>0</PostSeq>
   <Company>0A</Company>
   <SubmittedBy/>
   <SubmittedDate/>
  </ROW>
 </cdControl>

  <cdControlDefaultTaxCodes name="cdControlDefaultTaxCodes"
alias="cdControlDefaultTaxCodes" keys="Batch,TaxCode">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <TaxCode>Y15</TaxCode>
    <Seq>1</Seq>
   </ROW>
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <TaxCode>AGS</TaxCode>
    <Seq>2</Seq>
   </ROW>
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <TaxCode>PERC10</TaxCode>
    <Seq>3</Seq>
   </ROW>
```

```
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>PERC6</TaxCode>
      <Seq>4</Seq>
    </ROW>
  </cdControlDefaultTaxCodes>

 <cdMaster name="cdMaster" alias="cdMaster" keys="Batch,CheckNo">
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <CheckNo>000000567001</CheckNo>
    <TransDate>2013-08-27T00:00:00.000</TransDate>
    <BankCode>ASF Env.</BankCode>
    <Payee>MTC Payee</Payee>
    <Posted>N</Posted>
    <Seq>1</Seq>
    <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
    <CurrencyExchangeOverrideDate/>
    <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
    <Status>N</Status>
    <AuthorizedBy/>
    <RejectReason/>
    <ModUser/>
    <ModDate/>
  </ROW>
 </cdMaster>
 <cdDetail name="cdDetail" alias="cdDetail" keys="Batch,CheckNo,PKey">
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <CheckNo>000000567001</CheckNo>
    <PKey>MTCAPIPKey1</PKey>
    <Seq>1</Seq>
    <Description>Line 1</Description>
    <WBS1>A2005US210</WBS1>
    <WBS2/>
    <WBS3/>
```

```
      <Account>52101</Account>
      <Amount>1075</Amount>
      <NetAmount>1000</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <LinkCompany/>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <Seq>2</Seq>
      <Description>Line 2</Description>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
      <Account>51500</Account>
      <Amount>2290.12</Amount>
      <NetAmount>2000</NetAmount>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <LinkCompany/>
    </ROW>
  </cdDetail>

  <cdDetailTax name="cdDetailTax" alias="cdDetailTax"
keys="Batch,CheckNo,PKey,TaxCode">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>LOW3HIGH12</TaxCode>
      <TaxAmount>30</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
```

```
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>15</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>AGS</TaxCode>
      <TaxAmount>30</TaxAmount>
      <CompoundOnTaxCode>AHI</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>25</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>LOW4HIGH11</TaxCode>
      <TaxAmount>90</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
```

```
      <CheckNo>000000567001</CheckNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y11</TaxCode>
      <TaxAmount>100.12</TaxAmount>
      <CompoundOnTaxCode>NY_TAX</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
  </cdDetailTax>
 </REC>
</RECS>
```

Employee Expenses

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <REC>
  <exControl name="exControl" alias="exControl" keys="Batch">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <Recurring>Y</Recurring>
    <Posted>N</Posted>
    <Creator>MTC</Creator>
    <Period>201308</Period>
    <EndDate>2013-08-26T00:00:00.000</EndDate>
    <Total>0</Total>
    <PostPeriod>0</PostPeriod>
    <PostSeq>0</PostSeq>
    <AdvanceAmount>0</AdvanceAmount>
    <Company>0A</Company>
    <DefaultCurrencyCode>USD</DefaultCurrencyCode>
    <SubmittedBy/>
    <SubmittedDate/>
   </ROW>
  </exControl>

  <exControlDefaultTaxCodes name="exControlDefaultTaxCodes"
alias="exControlDefaultTaxCodes" keys="Batch,TaxCode">
   <ROW tranType="INSERT">
```

```
      <Batch>MTC-API</Batch>
      <TaxCode>Y15</TaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>AGS</TaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <TaxCode>PERC6</TaxCode>
      <Seq>3</Seq>
    </ROW>
  </exControlDefaultTaxCodes>

 <exMaster name="exMaster" alias="exMaster" keys="Batch,Employee">
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
    <Employee>00UBILL1</Employee>
    <ReportDate>2013-08-25T00:00:00.000</ReportDate>
    <ReportName>MTC EX</ReportName>
    <Posted>N</Posted>
    <Seq>1</Seq>
    <AdvanceAmount>0</AdvanceAmount>
    <BarCode/>
    <DefaultCurrencyCode>USD</DefaultCurrencyCode>
    <PaymentExchangeOverrideMethod>N</PaymentExchangeOverrideMethod>
    <PaymentExchangeOverrideDate/>
    <PaymentExchangeOverrideRate>0</PaymentExchangeOverrideRate>
    <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
    <CurrencyExchangeOverrideDate/>
    <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
    <Status>N</Status>
    <AuthorizedBy/>
```

```
      <RejectReason/>
      <ModUser/>
      <ModDate/>
    </ROW>
  </exMaster>
  <exDetail name="exDetail" alias="exDetail" keys="Batch,MasterPKey,PKey">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <Seq>1</Seq>
      <TransDate>2013-08-18T00:00:00.000</TransDate>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
      <Account>62101</Account>
      <Amount>1075</Amount>
      <Description>Meal</Description>
      <SuppressBill>N</SuppressBill>
      <NetAmount>1000</NetAmount>
      <PaymentExchangeRate>0</PaymentExchangeRate>
      <PaymentAmount>1075</PaymentAmount>
      <PaymentExchangeInfo/>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <CurrencyCode>USD</CurrencyCode>
      <OriginatingVendor/>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <Seq>2</Seq>
      <TransDate>2013-08-18T00:00:00.000</TransDate>
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
```

```
      <Account>62100</Account>
      <Amount>2300.12</Amount>
      <Description>Hotel</Description>
      <SuppressBill>N</SuppressBill>
      <NetAmount>2000</NetAmount>
      <PaymentExchangeRate>0</PaymentExchangeRate>
      <PaymentAmount>2300.12</PaymentAmount>
      <PaymentExchangeInfo/>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <CurrencyCode>USD</CurrencyCode>
      <OriginatingVendor/>
    </ROW>
  </exDetail>

  <exDetailTax name="exDetailTax" alias="exDetailTax"
keys="Batch,MasterPKey,PKey,TaxCode">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>LOW3HIGH12</TaxCode>
      <TaxAmount>30</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>15</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
```

```
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>AGS</TaxCode>
      <TaxAmount>30</TaxAmount>
      <CompoundOnTaxCode>AHI</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxAmount>25</TaxAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>LOW4HIGH11</TaxCode>
      <TaxAmount>100</TaxAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <MasterPKey>MTCAPIMasterPKey1</MasterPKey>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y11</TaxCode>
      <TaxAmount>100.12</TaxAmount>
      <CompoundOnTaxCode>NY_TAX</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
  </exDetailTax>
 </REC>
```

```
</RECS>
```

Journal Entries

```
<RECS xmlns="http://deltek.vision.com/XMLSchema"
xmlns:xdv="http://deltek.vision.com/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <REC>
  <jeControl name="jeControl" alias="jeControl" keys="Batch">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <Recurring>Y</Recurring>
    <Posted>N</Posted>
    <Creator>MTC</Creator>
    <Period>201308</Period>
    <EndDate>2013-08-26T00:00:00.000</EndDate>
    <Total>115</Total>
    <PostPeriod>0</PostPeriod>
    <PostSeq>0</PostSeq>
    <Company>0A</Company>
    <DefaultCurrencyCode>USD</DefaultCurrencyCode>
    <SubmittedBy>00UBILL1</SubmittedBy>
    <SubmittedDate/>
    <AllowTaxEntries>Y</AllowTaxEntries>
   </ROW>
  </jeControl>
  <jeMaster name="jeMaster" alias="jeMaster" keys="Batch,RefNo">
   <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <RefNo>000000112201</RefNo>
    <TransDate>2013-08-26T00:00:00.000</TransDate>
    <Type>3</Type>
    <Description>MTC API JE</Description>
    <Posted>N</Posted>
    <Seq>1</Seq>
    <UpdateBilled>N</UpdateBilled>
    <CurrencyCode>USD</CurrencyCode>
    <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
```

```
    <CurrencyExchangeOverrideDate/>
    <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
    <Status>A</Status>
    <AuthorizedBy>00UBILL1</AuthorizedBy>
    <RejectReason/>
    <ModUser/>
    <ModDate/>
  </ROW>
</jeMaster>
<jeDetail name="jeDetail" alias="jeDetail" keys="Batch,RefNo,PKey">
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <RefNo>000000112201</RefNo>
    <PKey>MTCAPIPKey1</PKey>
    <Seq>1</Seq>
    <WBS1>TEST</WBS1>
    <WBS2>310</WBS2>
    <WBS3/>
    <Account>52300</Account>
    <DebitAmount>0</DebitAmount>
    <CreditAmount>100</CreditAmount>
    <Description>Line 2 - Credit</Description>
    <TaxType>O</TaxType>
    <SuppressBill>N</SuppressBill>
    <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
    <LinkCompany/>
    <DebitAmountFunctionalCurrency>0</DebitAmountFunctionalCurrency>
    <CreditAmountFunctionalCurrency>0</CreditAmountFunctionalCurrency>
    <DebitAmountProjectCurrency>0</DebitAmountProjectCurrency>
    <CreditAmountProjectCurrency>0</CreditAmountProjectCurrency>
  </ROW>
  <ROW tranType="INSERT">
    <Batch>MTC-API</Batch>
    <RefNo>000000112201</RefNo>
    <PKey>MTCAPIPKey2</PKey>
    <Seq>2</Seq>
```

```
      <WBS1>A2005US210</WBS1>
      <WBS2/>
      <WBS3/>
      <Account>52101</Account>
      <DebitAmount>100</DebitAmount>
      <CreditAmount>0</CreditAmount>
      <Description>Line 1 - Debit</Description>
      <TaxType>I</TaxType>
      <SuppressBill>N</SuppressBill>
      <CurrencyExchangeOverrideRate>0</CurrencyExchangeOverrideRate>
      <LinkCompany/>
      <DebitAmountFunctionalCurrency>0</DebitAmountFunctionalCurrency>
      <CreditAmountFunctionalCurrency>0</CreditAmountFunctionalCurrency>
      <DebitAmountProjectCurrency>0</DebitAmountProjectCurrency>
      <CreditAmountProjectCurrency>0</CreditAmountProjectCurrency>
    </ROW>
  </jeDetail>

  <jeDetailTax name="jeDetailTax" alias="jeDetailTax" keys="Batch,RefNo,PKey,TaxCode">
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>LOW3HIGH12</TaxCode>
      <TaxDebitAmount>0</TaxDebitAmount>
      <TaxCreditAmount>3</TaxCreditAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxDebitAmount>0</TaxDebitAmount>
      <TaxCreditAmount>30</TaxCreditAmount>
```

```
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
      <PKey>MTCAPIPKey1</PKey>
      <TaxCode>AGS</TaxCode>
      <TaxDebitAmount>0</TaxDebitAmount>
      <TaxCreditAmount>30</TaxCreditAmount>
      <CompoundOnTaxCode>AHI</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y15</TaxCode>
      <TaxDebitAmount>0</TaxDebitAmount>
      <TaxCreditAmount>30</TaxCreditAmount>
      <CompoundOnTaxCode>Y11</CompoundOnTaxCode>
      <Seq>1</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>LOW4HIGH11</TaxCode>
      <TaxDebitAmount>11</TaxDebitAmount>
      <TaxCreditAmount>0</TaxCreditAmount>
      <CompoundOnTaxCode></CompoundOnTaxCode>
      <Seq>2</Seq>
    </ROW>
    <ROW tranType="INSERT">
      <Batch>MTC-API</Batch>
      <RefNo>000000112201</RefNo>
```

```
      <PKey>MTCAPIPKey2</PKey>
      <TaxCode>Y11</TaxCode>
      <TaxDebitAmount>0</TaxDebitAmount>
      <TaxCreditAmount>30</TaxCreditAmount>
      <CompoundOnTaxCode>NY_TAX</CompoundOnTaxCode>
      <Seq>3</Seq>
    </ROW>
  </jeDetailTax>
 </REC>
</RECS>
```

Obtaining Schema Files for Transaction Areas

The latest schema files for Vision Web APIs can be obtained by using a Web service that returns schema file content (String) and the other by using Vision.

Using API

Use the following API to obtain schema file content for an Info Center.

GetSchema(ConnInfoXML, TransactionArea)	
Description	Based on the value for 'TransactionArea' parameter for the call, content of the schema file respective to the transaction area is returned back.

Use the following parameter values to retrieve schema files for various transaction areas:

Parameter Value	Transaction Area
APVouchers_Transaction	A/P Vouchers
APDisbursements_Transaction	A/P Disbursements
CashDisb_Transaction	Cash Disbursements
CashReceipts_Transaction	Cash Receipts
EmpExpense_Transaction	Employee Expenses
EmpRepayment_Transaction	Employee Repayments
Invoice_Transaction	Invoices
JournalEntry_Transaction	Journal Entry
LaborAdjust_Transaction	Labor Adjustments
Misc_Transaction	Miscellaneous Expenses

Parameter Value	Transaction Area
PrintsAndRepro_Transaction	Prints and Reproductions
Timesheet_Transaction	Timesheets
Unit_Transaction	Units
UnitByProject_Transaction	Units by project

List of Vision Transaction Web APIs

Each API for transactions takes two parameters: One is the XML string with connection information (ConnXML), and the other is the XML string that contains the transaction data.

A/P Disbursements
AddAPDisbursementsTransaction(ConnXML, DataXML)
UpdateAPDisbursementsTransaction(ConnXML, DataXML)
DeleteAPDisbursementsTransaction(ConnXML, DataXML)
A/P Vouchers
AddAPVouchersTransaction(ConnXML, DataXML)
UpdateAPVouchersTransaction(ConnXML, DataXML)
DeleteAPVouchersTransaction(ConnXML, DataXML)
Cash Disbursements
AddCashDisbTransaction(ConnXML, DataXML)
UpdateCashDisbTransaction(ConnXML, DataXML)
DeleteCashDisbTransaction(ConnXML, DataXML)
Cash Receipts
AddCashReceiptsTransaction(ConnXML, DataXML)
UpdateCashReceiptsTransaction(ConnXML, DataXML)
DeleteCashReceiptsTransaction(ConnXML, DataXML)
Employee Expenses
AddEmpExpenseTransaction(ConnXML, DataXML)

UpdateEmpExpenseTransaction(ConnXML, DataXML)
DeleteEmpExpenseTransaction(ConnXML, DataXML)
Employee Repayments
AddEmpRepaymentTransaction(ConnXML, DataXML)
UpdateEmpRepaymentTransaction(ConnXML, DataXML)
DeleteEmpRepaymentTransaction(ConnXML, DataXML)
Invoices
AddInvoiceTransaction(ConnXML, DataXML)
UpdateInvoiceTransaction(ConnXML, DataXML)
DeleteInvoiceTransaction(ConnXML, DataXML)
Journal Entries
AddJournalEntryTransaction(ConnXML, DataXML)
UpdateJournalEntryTransaction(ConnXML, DataXML)
DeleteJournalEntryTransaction(ConnXML, DataXML)
Labor Adjustments
AddLaborAdjustTransaction(ConnXML, DataXML)
UpdateLaborAdjustTransaction(ConnXML, DataXML)
DeleteLaborAdjustTransaction(ConnXML, DataXML)
Miscellaneous Expenses
AddMiscTransaction(ConnXML, DataXML)
UpdateMiscTransaction(ConnXML, DataXML)
DeleteMiscTransaction(ConnXML, DataXML)
Prints and Reproductions
AddPrintsReproTransaction(ConnXML, DataXML)
UpdatePrintsReproTransaction(ConnXML, DataXML)
DeletePrintsReproTransaction(ConnXML, DataXML)

Timesheets
AddTimesheetTransaction(ConnXML, DataXML)
UpdateTimesheetTransaction(ConnXML, DataXML)
DeleteTimesheetTransaction(ConnXML, DataXML)
Units
AddUnitTransaction(ConnXML, DataXML)
UpdateUnitTransaction(ConnXML, DataXML)
DeleteUnitTransaction(ConnXML, DataXML)
Units by Project
AddUnitByProjectTransaction(ConnXML, DataXML)
UpdateUnitByProjectTransaction(ConnXML, DataXML)
DeleteUnitByProjectTransaction(ConnXML, DataXML)

Web APIs to Post Transactions

A single Web API is used to post transaction data. The API to post transactions takes the following parameters:

- **ConnXML** — This is connection information in XML format.
- **TransType** — This is the type of accounting transaction data that is being posted (possible values are listed below).
- **BatchList** — This is the list of transaction files (identified by the batch number). If there are multiple values, they must be separated by commas.
- **Period** — This is the accounting period (as an integer) for which the transaction needs to be posted.

Values for TransType Parameter

Value	Transaction Type
AP	A/P Disbursements and A/P Vouchers
CD	Cash Disbursements
CR	Cash Receipts
ER	Employee Repayments
EX	Employee Expenses

Value	Transaction Type
IN	Invoices
JE	Journal Entries
LA	Labor Adjustments
MI	Miscellaneous Expenses
PR	Print and Reproductions
TS	Timesheets
UN	Units
UP	Units by Project

PostTransaction(ConnXML, TransType, BatchList, Period)

Transaction Security Validation

In addition to validation of login and access to transaction application area, the following security rights are validated and used when processing transaction Web API calls. These security options are in Vision on the Accounting tab in **Configuration » Security » Roles**.

- **Record level security** — When the **Apply to All Transaction Center** option is selected, record-level security as specified for Info Centers is validated while processing transaction data. If any of the transactions sent through the API references Info Center records (for example, projects) that the Vision login role does not have access to in the Info Center, those transaction entries will not be saved to the database.
- **Access to transaction types** — If the **Full access to all transaction types** option is not selected, access to transaction types as listed in the transaction-type grid is validated. Transaction data that is sent through APIs is processed and saved to the database only if the Vision login (used in ConnXML parameter for Web APIs) has access to respective transaction-types.
- **Transactions in closed periods** — When the **Allow Processing in Closed Periods** option is selected, the posting Web API allows postings in closed periods.
- **Transactions in prior periods** — When the **Allow Processing in Prior Periods** option is selected, the posting Web API allows postings in prior periods. The current period is the most recent period that has been opened and periods before the current period are considered as prior periods.

Transaction Data Validation

Vision accounting applications employ several data validation processes to validate data that is entered by the user and these validations are done either during the data entry or at the time of Save. All of the business logic that constitute data validation that is done during and after data entry in Vision, is also applicable when processing Web API calls to add/update/post transactions. These data validations include checks with respect to referential dependencies, record-level security, format of the data, data integrity, and access rights. Given the enormity of

the details of specific data validation rules and logic, they are not individually listed in this section. The sections of application-specific online help may be used to understand validation rules in each of the accounting transaction areas.

Multicompany Support for Transactions

In Vision databases where the Multicompany feature has been activated, the Web API calls to manipulate accounting transaction information are processed in the context of the company that is used in a given transaction. The company information included in XML for the Control entity (with reference to schema for a transaction-type) is used as the company for which the transaction is being processed and any business logic or data validation that is dependent on the company is processed in the context of the company information specified. For Vision databases where the Multicompany feature is not in use, the company information in Control entity (with reference to the schema) can be set to single-space.

Transaction API Return Messages

The return messages in XML for transaction Web APIs follow the standard error message format applicable to all Web APIs for Vision in general.

Return Messages

The following table contains sample return messages with respect to various events and errors.

Event/Error and Error Code	Return message in XML

Successful API call
Return Code: 1
	`<DLTKVisionMessage>`
	`<ReturnCode>1</ReturnCode>`
	`<ReturnDesc>Service has been successfully run</ReturnDesc>`
	`</Detail>`
	`</DLTKVisionMessage>`

Schema validation error
Return Code:
ErrSchemaVal
	`<DLTKVisionMessage>`
	`<ReturnCode>ErrSchemaVal</ReturnCode>`
	`<ReturnDesc>Schema Validation Error</ReturnDesc>`
	`<Detail>Validation error on line 1 col 490 The 'http://deltek.vision.com/XMLSchema:suffix' element has an invalid value according to its data type. An error occurred at , (1, 490). ... Continuing Validation ...`
	`</Detail>`
	`</DLTKVisionMessage>`

Vision login validation error
Return Code:
ErrLoginVal
	`<DLTKVisionMessage>`
	`<ReturnCode>ErrLoginVal</ReturnCode>`
	`<ReturnDesc>Login Validation Error</ReturnDesc>`
	`<Detail>Vision login failed, no data is saved to the database</Detail>`
	`</DLTKVisionMessage>`

Event/Error and Error Code	Return message in XML
Vision record access rights error **Return Code:** **ErrRecAccess**	<DLTKVisionMessage> <ReturnCode>ErrRecAccess</ReturnCode> <ReturnDesc>Record Access Validation Error</ReturnDesc> <Detail> <AddRec> <Message>Permission to add new records is denied</Message> </AddRec> <UpdateRec> <Message>Permission to update records is denied</Message> <ReturnValues> <keyVal>Admin162326539399</keyVal> </ReturnValues> </UpdateRec> </Detail> </DLTKVisionMessage>

Event/Error and Error Code	Return message in XML
Vision data validation error **Return Code:** **ErrDataVal**	<DLTKVisionMessage> <ReturnCode>ErrDataVal</ReturnCode> <ReturnDesc>Data Validation Error</ReturnDesc> <Detail> <ValidationResult> <Message> <Table>leads</Table> <Column>status</Column> <ColValue>sysNew1</ColValue> <RowNum>1</RowNum> <PrimaryRecKeyVal> Admin162326539399</PrimaryRecKeyVal> <MsgCode>VAL1001</MsgCode> <MsgDesc>Value not in the master/code table!</MsgDesc> </Message> <Message> <Table>leads</Table> <Column>employee</Column> <ColValue>0002011</ColValue> <RowNum>2</RowNum> <PrimaryRecKeyVal> Admin162326539399</PrimaryRecKeyVal> <MsgCode>VAL1001</MsgCode> <MsgDesc>Value not in the master/code table!</MsgDesc> </Message> </ValidationResult> </Detail> </DLTKVisionMessage>
Save error (record exists) **Return Code:** **ErrSave**	<DLTKVisionMessage> <ReturnCode>ErrSave</ReturnCode> <ReturnDesc>Save Error</ReturnDesc> <Detail>Record already exists and cannot be added</Detail> </DLTKVisionMessage>

Event/Error and Error Code	Return message in XML
Add records (using 'Add' APIs) **Return Code: 1** (when successful)	<DLTKVisionMessage> <ReturnCode>1</ReturnCode> <ReturnDesc>Records Added</ReturnDesc> <Detail> <ReturnValues> <AddRec> <Row Number="1"> <KeyVal>Admin162326539399</KeyVal> </Row> <Row Number="2"> <KeyVal>Admin162326539799</KeyVal> </Row> </AddRec> </ReturnValues> </Detail> </DLTKVisionMessage>
Schema generation error **Return Code:** **ErrGenSchema**	<DLTKVisionMessage> <ReturnCode>ErrGenSchema</ReturnCode> <ReturnDesc>Custom Tabs Schema Generation Error</ReturnDesc> <Detail></Detail> </DLTKVisionMessage>
Get schema from server error **Return Code:** **ErrGetSchema**	<DLTKVisionMessage> <ReturnCode>ErrGetSchema</ReturnCode> <ReturnDesc>Get Schema File - Error</ReturnDesc> <Detail></Detail> </DLTKVisionMessage>
Error when retrieving records **Return Code:** **ErrGetRecs**	<DLTKVisionMessage> <ReturnCode>ErrGetRecs</ReturnCode> <ReturnDesc>Get Records - Data Retrieval Error</ReturnDesc> <Detail></Detail> </DLTKVisionMessage>

Event/Error and Error Code	Return message in XML
Error when validating data sent for deleting records **Return Code:** **ErrDeleteVal**	<DLTKVisionMessage> <ReturnCode>ErrDeleteVal</ReturnCode> <ReturnDesc>Delete Records - Input XML Error</ReturnDesc> <Detail></Detail> </DLTKVisionMessage>
Error when deleting records **Return Code:** **ErrDeleteRecs**	<DLTKVisionMessage> <ReturnCode>ErrDeleteRecs</ReturnCode> <ReturnDesc>Delete Records - Save Error</ReturnDesc> <Detail></Detail> </DLTKVisionMessage>

Integrating Web Services/API into an Application

You can use one of the following three ways to access Vision Web services from an external application, depending on the tool that you use for application development:

- **Using WSDL file generated for VB** — You can add the Deltek.Vision.WebServiceAPI.Server.dll file to a VB.Net project. The Web services that are listed in this document can be called using an instance of "Deltek.Vision.WebServiceAPI."

  The following is a sample excerpt of the code:

```
Dim returnMessage As String
Dim ConnInfoXML As String
Dim DataXML As String
Dim InfoCenter As String


‘<Set values for variables here!>


Dim VisionWS As New Deltek.Vision.WebServiceAPI


returnMessage = VisionWS.SendDataToDeltekVision(InfoCenter, ConnInfoXML,
DataXML)


‘<Parse and process returnMessage here!>
```

- **Using WSDL file generated for C# language** — Use the Deltek.Vision.WebServiceAPI.dll file with projects that are developed in C#.
- **Using a compiled DLL with the application** — Use the WSDL file for VB or C# to compile a DLL. It can also be used with the calling application.

Security and Calls to Web Services

The Web accessible APIs (Web services) for Vision are accessed through the main VisionWS.asmx page deployed in <*InstallDir*>\ Vision\Web directory (where <*InstallDir*> is the directory where Deltek Vision is installed). The URL to this service page is embedded in the WSDL file. The calls to Web services use SOAP as the application-level protocol and use http as the transport protocol. The URL to the main service page can be found in the 'Constructor' in WSDL file.

For example, it would look like the following:

```
Public Sub New()
        MyBase.New
        Me.Url = "http://VisionServer/Vision/VisionWS.asmx"
End Sub
```

Since parameters for Web services are sent as plain text, there is an option to make secured calls to Web services using https instead of http. As long as the Web server supports secured https calls, Vision Web services can be accessed securely. In order to do it, change the URL above to use **https** instead of http.

Invoking a Web Service to Process Workflow Actions

The Workflow feature in Vision offers the capability to set up a wide range of workflow criteria and corresponding actions. In order to extend the ability to define actions whose processing logic is beyond the scope of standard actions available in the application, and specifically if actions involve dealing with data or applications outside of Vision, Vision provides the ability to invoke a Web service when a set of workflow conditions are met.

The guidelines for using a Web service with workflows, along with sample code snippets, are included in this section.

Invoke Web Service Versus Invoke Custom Method Options in Workflow

It is important to note the differences between invoking the Web services option and invoking the custom methods option for workflow. The use of Web services in workflow is suitable only under specific conditions.

When to Use Web Services

- Web services offer a standard way to leverage capabilities of different distributed applications that offer Web services, regardless of how they are built and where they are located.
- In the context of Vision, Web services are particularly suitable if a specific action needs to be triggered outside of Vision based on a specific event or set of events in Vision.
- Use the Web services option if existing Web services offered by others vendors and applications need to be invoked based on workflow events in Vision.
- The Web service option is also applicable when the code that captures custom logic to handle Vision events cannot be on the Vision server and needs to reside at a different location on the Intranet or the Internet.
- If the action for a workflow event needs to access/modify data in Vision database, it is strongly recommended that you use Invoke Custom Method rather than Invoke Web Service. Using a Web service to access a Vision database will not allow your queries to be part of the Vision database transaction.

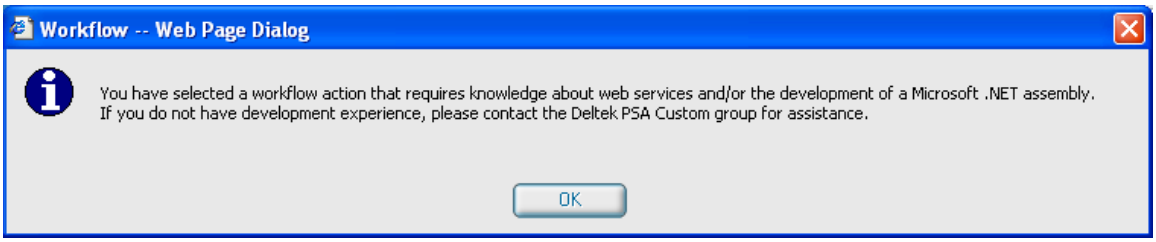
Advantages of Using a Custom Method

- Using a custom method from a custom DLL file residing on the Vision application server has several advantages. For more technical details, see the "Invoking a Custom Method to Process Workflow Actions" section on page 78 of this guide.
- The custom component/DLL used to invoke a custom method in workflow can inherit from Vision workflow API DLL, and the custom method can access some of the internal methods of Vision, which includes methods to access Vision database. This is certainly an advantage, and it also makes it easy to write code for the custom method.
- Any SQL access that is included in the custom method is handled as a part of Vision transaction.
- If the Save transaction fails in Vision, everything is rolled back, including what was processed in the custom method.

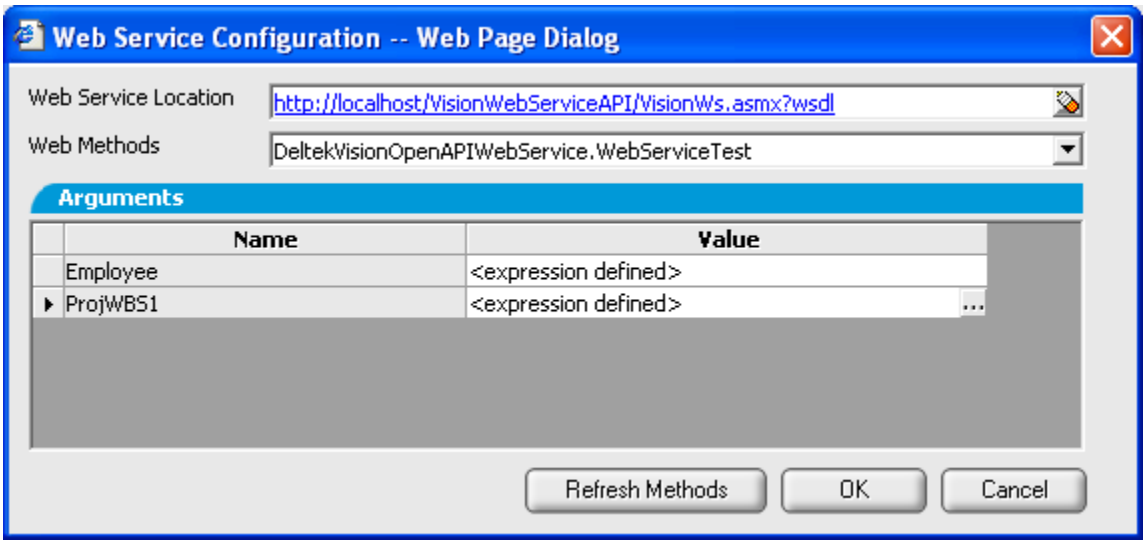
- Since a Vision transaction is already open and available for use, there is no need to use a connection string to connect to Vision database from the custom method.
- Since the custom component (DLL) that consists of the custom method is local, it offers better performance, and avoids serialization and de-serialization of data, which would be the case when remote Web services are used.
- The deployment is extremely simple because it involves copying of a single DLL file to a specified location on the Vision server.

Implementation Details

The **Insert** hyperlink on the **Actions** grid in the Workflow area provides a list of action types to choose from, including **Invoke Web Service**. When you select this option, Vision displays the following Workflow warning message.



When you click **OK** on this Workflow dialog box, the following Web Service Configuration dialog box displays. Enter required information about the particular Web service and the Web method to be invoked and the parameters that need to be passed to the Web method from Vision.



Fields on the Web Service Configuration Dialog Box

The following table describes the fields on the Web Service Configuration dialog box.

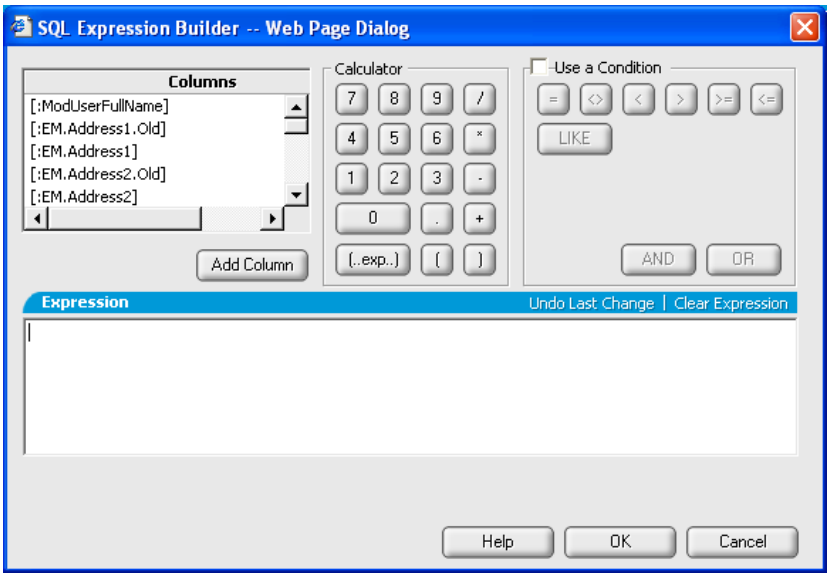
Field	Description
Web Service Location	This can be either a location or a file path for the WSDL file generated for the Web service. Alternatively, it can be a URL to the WSDL file located on a Web server. For Web services written in ASP.NET, it can also be a URL to the Web service 'asmx' page with "?WSDL" added at the end. This parameter generates WSDL information automatically. The following are some examples: WSDL File Location: \\Serv1\Share1\Proj1\VisionWorkflowWS.wsdl WSDL File URL: http://Serv1/VisionWorkflowWS/ VisionWorkflowWS.wsdl ASMX page URL: http://Serv1/ VisionWorkflowWS / VisionWfWS.asmx?wsdl
Web Method	Based on the information contained in the WSDL file, a list of available Web methods in the Web service prefills in the **Web Method** drop-down list. From the list of Web methods, select an appropriate method for Vision workflow.
Arguments	▪ The **Arguments** grid prefills with a list of arguments or parameters defined for the Web method you selected. Each argument's name is listed on the left column of the grid. The value for each of the arguments is populated in the right column of the grid. The values for arguments must be provided using the SQL Expression Builder.

SQL Expression Builder

The SQL Expression Builder is a powerful tool that allows you to build simple or complex SQL expressions using Vision database fields. The expression can be as simple as the column name from a table in the database, or it can be a complex SQL statement with various operators and conditions. The list of columns available for use is based on the workflow table selected for workflow. When the main record/table for the Info Center is selected as the workflow table, the list of columns includes columns from the main Info Center table and also any form-type or grid-type user-defined fields created for the Info Center.

All character/string type fields are added to the expression with qualifiers (single quotes) around them when you click the **Add Column** button. All numeric type fields are added with no qualifiers.

The SQL expressions are not validated or evaluated when they are being built. Deltek recommends that you check expressions for accuracy before you save them.



Refreshing Modified Values

If the currently referenced Web service has been modified to include new Web methods or if existing Web methods have been altered with respect to the list of arguments, then you can use the **Refresh Methods** button on the Web Service Configuration dialog box. When you click this button, it forces Vision to read from the latest WSDL file generated for the Web service. The Web method drop-down list is refreshed with the latest values. If the list of arguments has changed for the Web method that is currently selected, the **Arguments** grid is automatically refreshed to reflect the change.

Application Development Notes

The Web service is run when a record in the workflow table is saved in Vision. When the Web service is run, the transaction may be still pending depending on whether or not there are other tables that need be saved by Vision as part of the same transaction.

The processing of Web service is independent of successful completion of a‘Save’ transaction in Vision. If a ‘Save’ transaction fails because of problems encountered while saving changes to other tables (after processing changes for the workflow table), the Save transaction is rolled back in Vision. However the Web service would have run, and it cannot be rolled back.

The Web service (Web method) can return an XML string back to Vision so that a message can be displayed in Vision. The message can be a warning type or it can be an error. The format for return XML message is detailed in the following example. If an empty string or empty error XML is returned, it means that there were no errors or warnings when the Web service was run.

Error Message

```
<errors>
      <error>Error Message 1</error>
      <error>Error Message 2</error>
      <error>Error Message 3</error>
</errors>
```

Error messages 1, 2, and 3 are actual messages that display in Vision.

Warning Message

Returned message can also be designated as warnings rather than as errors. An XML attribute ‘warning’ is used to indicate that the message returned is to be displayed as a warning in Vision. The XML format for warning messages is exactly the same as the format used for error messages, but an extra attribute is used, as shown in the following example.

```
<errors warning=”y”>
      <error>Warning Message 1</error>
      <error>Warning Message 2</error>
      <error>Warning Message 3</error>
</errors>
```

Return Message to Indicate Successful Running of the Web Service

An empty string ('') or the following empty XML error message is returned.

```
<errors></errors>
```

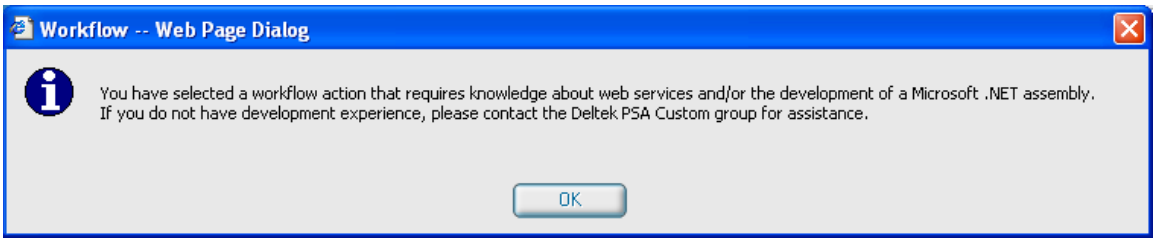
Invoking a Custom Method to Process Vision Workflow Actions

The Workflow feature in Vision offers the capability to set up a wide range of workflow criteria and corresponding actions. However, there are practical constraints to implementing all combinations of workflow criteria and actions both within and across Info Centers. In order to extend the ability to define actions whose processing logic is beyond the scope of standard actions available in the application, Vision provides two ways of designating customized actions to a set of workflow conditions. One option, which is described in this section, is to invoke a custom method from an external component or DLL, developed outside of Vision. This option offers flexibility and convenience to customize the business logic for actions that need to be initiated when a particular set of workflow conditions are met.

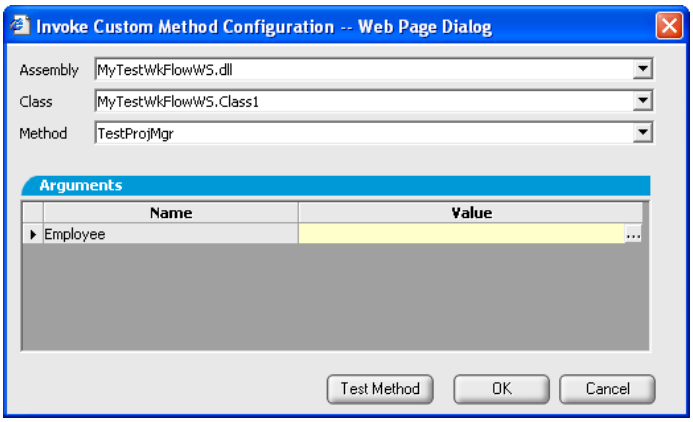
To ensure that a custom method when invoked through workflow works properly, there are guidelines as to how the custom component (DLL) needs to be developed. It is required that the DLL consisting of the custom method is developed using Microsoft.Net platform. The guidelines for developing a custom component or an assembly, along with sample code snippets, are included in this section. The code excerpts used for examples below are in VB.Net.

Implementation Details

The **Insert** hyperlink on the **Actions** grid on the Workflow form displays a list of action types to choose from, including **Invoke Custom Method**. When you select this option, Vision displays the following warning message.



When you click **OK**, the following dialog box displays to collect the required information about the particular custom method to be invoked and the parameters that must be passed to the custom method from Vision.



Fields on the Invoke Custom Method Configuration Dialog Box

The following table describes the fields on the Invoke Custom Method Configuration dialog box.

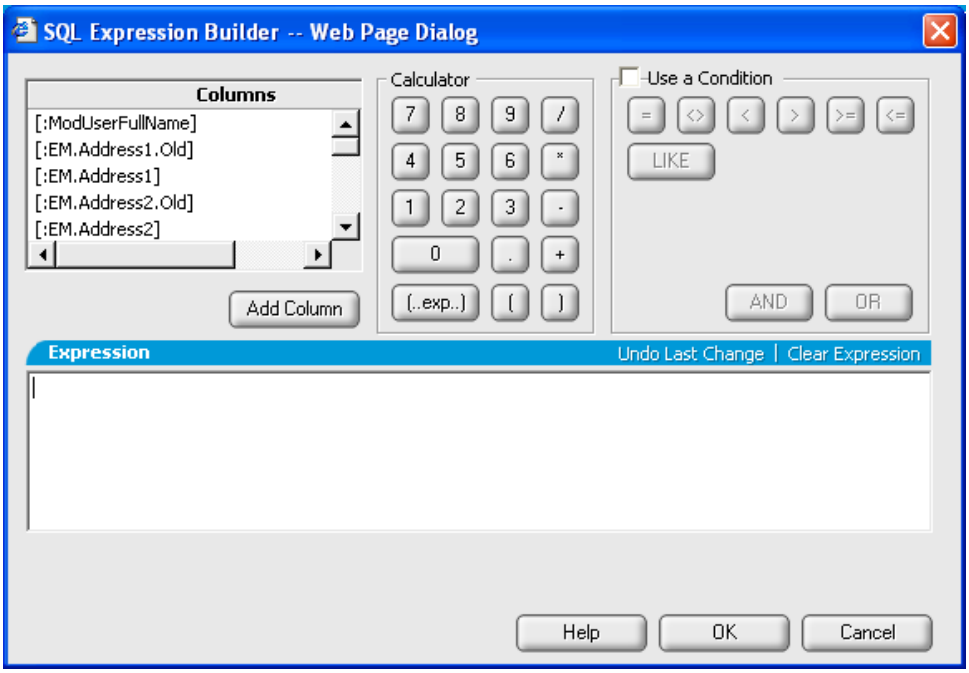
Field	Description
Assembly	This is the name of the actual Microsoft.Net assembly or the DLL that consists of the custom method that is being invoked. The drop-down list is automatically populated based on Microsoft.Net assemblies or DLLs that are present in <*VisionInstallDir*>\Workflow folder. Assemblies or DLL files that are used for invoking custom methods must be placed in the <*VisionInstallDir*>/Workflow folder. Vision does not provide the option to use an assembly that is located elsewhere.
Class	When you select a specific assembly from the list, Vision automatically reads the assembly and prefills the **Class** field drop-down list with classes available in the assembly.
Method	The drop-down list in the **Method** field is prefilled with a list of methods (functions or subs) that are available in the selected class. The method that you select from the list is the method that will be invoked when workflow conditions are met.
Arguments	The **Arguments** grid is prefilled with a list of arguments or parameters defined for the method you select. Each argument's name is listed on the left column of the grid. The value for each of the arguments is populated in the right column of the grid. The values for arguments must be provided using SQL Expression Builder.

The SQL Expression Builder

The SQL Expression builder is a powerful tool that enables you to build simple or complex SQL expressions using Vision database fields. The expression can be as simple as the column name from a table in the database, or it can be a complex SQL statement with various operators and conditions. The list of columns available for use is based on the workflow table selected for workflow. When you select the main record/table for the Info Center as the workflow table, the list of columns includes columns from the main Info Center table and also any form-type or grid-type user-defined fields created for the Info Center.

All character/string type fields are added to the expression with qualifiers (single quotes) around them when you click the **Add Column** button. All numeric type fields are added with no qualifiers.

The SQL expressions are not validated or evaluated when they are being built. Deltek recommends that you check expressions for accuracy before you save them.



Testing the Method

After you enter all of the configuration information, click the **Test Method** button on the Invoke Custom Method Configuration dialog box to test whether or not the specified custom method can be successfully invoked. Regardless of the SQL expressions defined for the arguments, the test method requires absolute values for all arguments to perform the test.

Application Development Notes

The descriptions and code snippets in the following table provide details about creating an assembly that consists of the custom method invoked from workflow. The development platform used is Microsoft.Net and the code examples are in VB.Net.

After a new project (typically of Class-Library type) is created, add 'Deltek.Vision.WorkflowAPI.Server.dll' to references. All of the assemblies or DLLs used by Vision are found in *<Vision-Install-Dir>\web\bin* directory. This DLL provides access to some of the workflow APIs that can be called from the custom method. It also provides a standard way (consistent with what is done in Vision) to return errors or messages from custom method back to users.

Deltek.VisionWorkflowAPI.Server.dll has two classes:

- WorkflowBaseClass
- WorkflowAPIMethods

WorkflowBaseClass

The WorkflowBaseClass consists of the following functions and subs that can be called from a custom method.

Name	Description
AddError	This adds an error message to display in Vision. This is used in situations to indicate that the error will abort the Save transaction, and all changes are rolled back. **Parameter**: Error Message As String **Returns**: N/A
AddFatal	This adds an error message to display as fatal in Vision. This is used in situations to indicate that the error will abort the Save transaction, and all changes are rolled back. Fatal errors are usually raised when .NET runtime exceptions are captured. **Parameter**: Error Message As String **Returns**: N/A
AddInformation	This adds a message to display as information in Vision. **Parameter**: Information Message as String **Returns**: N/A
AddWarning	This adds a message to display as warning in Vision. **Parameter**: Warning Message as String **Returns**: N/A
ExecuteSQL	This executes a SQL query on the database. It is generally used for Update or Insert statements where data is not expected in return. The SQL statement is executed as a part of the current transaction. **Parameter**: SQL Query as String **Returns**: Integer – Number of rows
QueryData	This executes a SQL query on the database and returns back data. It is generally used for Select statements where a set of records is expected in return. The SQL statement is executed as a part of the current transaction and eliminates the possibility of deadlocks in the database. **Parameter**: SQL Query as String **Returns**: DataTable (VB.Net)

WorkflowAPIMethods

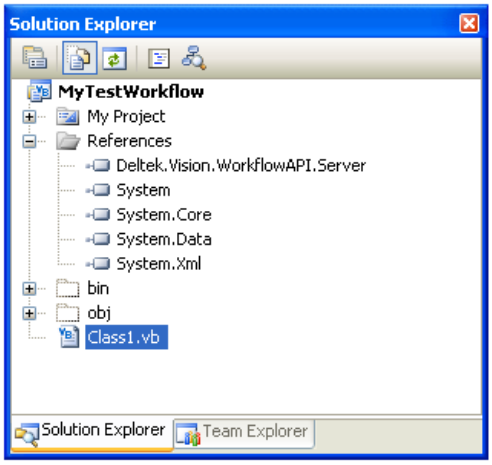
The WorkflowAPIMethods class consists of the following functions that can be called from a custom method. For details about parameters and return values, refer to the WorkflowCustomHelp.chm help file.

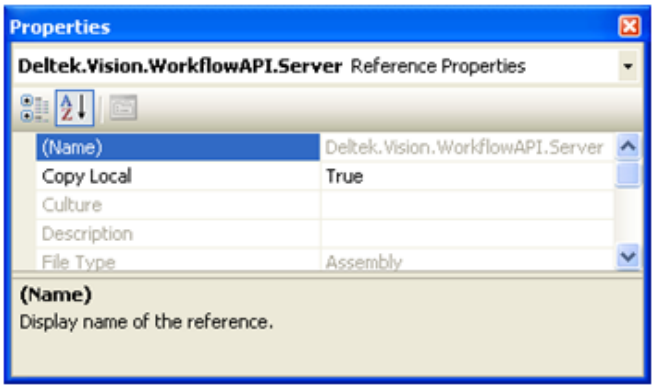
The naming convention used for Info Centers below is based on the default names used in Vision. These Info Center names may be different if you have changed them in your database.

Name	Description
CreateProjectFromExistingProject	This creates a new project by copying information from an existing project.
CreateProjectFromOpportunity	This creates a new project from an opportunity record.
CreateProjectFromPlan	This creates a new project from an existing project plan.
CreateProjectFromPromotionalProject	This creates a new project from an existing promotional project.
CreateProjectFromTemplate	This creates a new project using a project template.
CreatePromotionalProjectFromOpportunity	This creates a new promotional project from an existing opportunity record.

Illustrated Examples with Code

When the Deltek.Vision.WorkflowAPI.Server.dll file is added to references, select it in Solution Explorer, and make sure that the **Copy Local** property in the Properties window is set to **False**. This will stop the copying of the referenced DLL to the output directory, and then it does not complain about the dependencies of Deltek.Vision.WorkflowAPI.Server.dll when compiling your custom assembly.





Sample Code for the Custom Method

```
Imports Deltek.Vision.WorkflowAPI.Server

Public Class Class1
    Inherits WorkflowBaseClass

    Public Sub TestProjMgr(ByVal Employee As String)
        Dim sSQL As String
        Dim sPrjMgr As String
        Dim dtDataTable As DataTable

        If Employee.Length > 0 Then

            sSQL = "SELECT custPrjMgr FROM EmployeeCustomTabFields "
            sSQL += "WHERE employee = '" & Employee & "'"

            dtDataTable = QueryData(sSQL)

            If dtDataTable.Rows.Count < 1 Then
                AddError("No data returned from the custom query")
            Else
                sPrjMgr = dtDataTable.Rows(0).Item("custPrjMgr").ToString()
            End If

            If String.Compare(Trim(IsProjMgr), "PM", True) <> 0 Then
                AddError("Not a designated Project Manager!")
```

```
            End If
        End If
    End Sub
End Class
```

As shown in the previous example, the workflow API DLL “Deltek.Vision.WorkflowAPI.Server” is imported into the project and the class “Class1” that consists of the custom method to be used, is inherited from WorkflowBaseClass.

If other standard Vision functions available for workflow need to be used by the custom method, the class ‘WorkflowAPIMethods’ from Deltek.Vision.WorkflowAPI.Server is instantiated as shown in the following examples. Any applicable function from the class is called.

```
Dim bReturnValue As Boolean

Dim wfAPIMethods As New Deltek.Vision.WorkflowAPI.Server.WorkflowAPIMethods

bReturnValue = wfAPIMethods.CreateProjectFromOpportunity(......)
```

Extending Data Validation Business Logic for Timesheets

When timesheets are submitted in Vision, the data validation business logic can be extended to process custom validation rules using Web services. The custom validation process is treated as a part of a single “submit” transaction. In other words, based on the success or failure of the extended custom data validation, the entire submit operation is either completed or stopped.

The custom data validation process can return messages as errors or warnings in Vision. If no messages are returned, it is interpreted that the custom data validation was successful, and the timesheet is submitted and saved.

All of the custom data validation rules must be encapsulated in one single Web service, and one Web method is designated to be invoked when timesheets are submitted. In Vision, a timesheet configuration option is available to specify the Web service and the Web method that need to be invoked when timesheets are submitted.

The Web service that is associated with the timesheet-data validation can also be used to launch other events and services that are outside and independent of Vision. Currently, it is not possible to invoke any internal workflow processes in Vision.

Implementation Details

When the timesheet is submitted after processing standard data validation rules, the specified Web service is called, and the designated Web method is passed to the following parameter:

Parameter For The Web Method That Is Invoked When A Timesheet Is Submitted	
Timesheet Record (String - Entire timesheet record in XML format)	This is the entire timesheet record, which includes data from multiple timesheet-related tables in the Vision database represented in XML format.

Timesheet Record in XML

The entire timesheet record in XML format is passed to the Web method to process custom data validation business logic. A single timesheet record consists mainly of data from two tables in the database:

- **tkMaster** — This table consists of a single row with master information for the record.
- **tkDetail** — This table is comprised of timesheet details for each of the days in the timesheet period. The custom data validation logic can be structured to validate any of the pieces of data in either the master or detail table. The XML format of the timesheet record that is passed to the Web method is given below.

See the Vision Data Dictionary for descriptions of fields in these tables.

Example:

The yellow-highlighted text applies only if you have unit data. The blue-highlighted text applies only if you have break time data.

```
<ROOT>
 <tkMaster>
   <ROW>
     <Employee>0A0001</Employee>
     <EndDate>2013-09-30T00:00:00.000</EndDate>
     <Submitted>N</Submitted>
     <Selected>N</Selected>
     <SubmittedBy>001031</SubmittedBy>
     <ApprovedBy/>
     <TKGroup/>
   </ROW>
 </tkMaster>
 <tkDetail>
   <EndDate>2013-09-30T00:00:00-07:00</EndDate>
   <Employee>0A0001</Employee>
   <Seq>4</Seq>
   <TransDate>2013-09-16T00:00:00-07:00</TransDate>
   <Category/>
   <WBS1>00BUDGET.002</WBS1>
   <WBS2>000W</WBS2>
   <WBS3 xml:space="preserve"/>
   <LaborCode>R4:00V:0W</LaborCode>
   <RegHrs>1.5000</RegHrs>
   <OvtHrs>0.0000</OvtHrs>
   <SpecialOvtHrs>0.0000</SpecialOvtHrs>
   <BillCategory>14</BillCategory>
   <Locale>MA</Locale>
   <RegAmt>0.0000</RegAmt>
   <OvtAmt>0.0000</OvtAmt>
   <SpecialOvtAmt>0.0000</SpecialOvtAmt>
   <RegAmtProjectCurrency>0.0000</RegAmtProjectCurrency>
   <OvtAmtProjectCurrency>0.0000</OvtAmtProjectCurrency>
   <SpecialOvtAmtProjectCurrency>0.0000</SpecialOvtAmtProjectCurrency>
```

```
    <BillExt>0.0000</BillExt>
    <OvtPct>0.0000</OvtPct>
    <SpecialOvtPct>0.0000</SpecialOvtPct>
    <Rate>0.0000</Rate>
    <OvtRate>0.0000</OvtRate>
    <SpecialOvtRate>0.0000</SpecialOvtRate>
    <RateProjectCurrency>0.0000</RateProjectCurrency>
    <OvtRateProjectCurrency>0.0000</OvtRateProjectCurrency>

<SpecialOvtRateProjectCurrency>0.0000</SpecialOvtRateProjectCurrency>
    <StartDateTime>2013-09-16T10:30:00-07:00</StartDateTime>
    <EndDateTime>2013-09-16T12:00:00-07:00</EndDateTime>
  </tkDetail>
  <tkDetail>
    <EndDate>2013-09-30T00:00:00-07:00</EndDate>
    <Employee>0A0001</Employee>
    <Seq>4</Seq>
    <TransDate>2013-09-17T00:00:00-07:00</TransDate>
    <Category/>
    <WBS1>00BUDGET.002</WBS1>
    <WBS2>000W</WBS2>
    <WBS3 xml:space="preserve"/>
    <LaborCode>R4:00V:0W</LaborCode>
    <RegHrs>8.0000</RegHrs>
    <OvtHrs>0.0000</OvtHrs>
    <SpecialOvtHrs>0.0000</SpecialOvtHrs>
    <BillCategory>14</BillCategory>
    <Locale>MA</Locale>
    <RegAmt>0.0000</RegAmt>
    <OvtAmt>0.0000</OvtAmt>
    <SpecialOvtAmt>0.0000</SpecialOvtAmt>
    <RegAmtProjectCurrency>0.0000</RegAmtProjectCurrency>
    <OvtAmtProjectCurrency>0.0000</OvtAmtProjectCurrency>
    <SpecialOvtAmtProjectCurrency>0.0000</SpecialOvtAmtProjectCurrency>
    <BillExt>0.0000</BillExt>
    <OvtPct>0.0000</OvtPct>
    <SpecialOvtPct>0.0000</SpecialOvtPct>
```

```
    <Rate>0.0000</Rate>
    <OvtRate>0.0000</OvtRate>
    <SpecialOvtRate>0.0000</SpecialOvtRate>
    <RateProjectCurrency>0.0000</RateProjectCurrency>
    <OvtRateProjectCurrency>0.0000</OvtRateProjectCurrency>

<SpecialOvtRateProjectCurrency>0.0000</SpecialOvtRateProjectCurrency>
    <StartDateTime>2013-09-17T10:30:00-07:00</StartDateTime>
    <EndDateTime>2013-09-17T18:30:00-07:00</EndDateTime>
  </tkDetail>
  <tkDetail>
    <EndDate>2013-09-30T00:00:00-07:00</EndDate>
    <Employee>0A0001</Employee>
    <Seq>5</Seq>
    <TransDate>2013-09-18T00:00:00-07:00</TransDate>
    <Category/>
    <WBS1>00BUDGET.001</WBS1>
    <WBS2 xml:space="preserve"/>
    <WBS3 xml:space="preserve"/>
    <LaborCode>R2:00O:0P</LaborCode>
    <RegHrs>3.5000</RegHrs>
    <OvtHrs>0.0000</OvtHrs>
    <SpecialOvtHrs>0.0000</SpecialOvtHrs>
    <BillCategory>5</BillCategory>
    <Locale>MA</Locale>
    <RegAmt>0.0000</RegAmt>
    <OvtAmt>0.0000</OvtAmt>
    <SpecialOvtAmt>0.0000</SpecialOvtAmt>
    <RegAmtProjectCurrency>0.0000</RegAmtProjectCurrency>
    <OvtAmtProjectCurrency>0.0000</OvtAmtProjectCurrency>
    <SpecialOvtAmtProjectCurrency>0.0000</SpecialOvtAmtProjectCurrency>
    <BillExt>0.0000</BillExt>
    <OvtPct>0.0000</OvtPct>
    <SpecialOvtPct>0.0000</SpecialOvtPct>
    <Rate>0.0000</Rate>
    <OvtRate>0.0000</OvtRate>
    <SpecialOvtRate>0.0000</SpecialOvtRate>
```

```
    <RateProjectCurrency>0.0000</RateProjectCurrency>
    <OvtRateProjectCurrency>0.0000</OvtRateProjectCurrency>

<SpecialOvtRateProjectCurrency>0.0000</SpecialOvtRateProjectCurrency>
    <StartDateTime>2013-09-18T08:30:00-07:00</StartDateTime>
    <EndDateTime>2013-09-18T12:00:00-07:00</EndDateTime>
  </tkDetail>
  <tkUnitDetail>
    <EndDate>2013-09-30T00:00:00-07:00</EndDate>
    <Employee>0A0001</Employee>
    <Seq>1</Seq>
    <TransDate>2013-09-16T00:00:00-07:00</TransDate>
    <WBS1>00BUDGET.001</WBS1>
    <WBS2 xml:space="preserve"/>
    <WBS3 xml:space="preserve"/>
    <UnitTable>9701300</UnitTable>
    <Unit>A0302</Unit>
    <Qty>10.0000</Qty>
    <PKey>TSAUDIT435861514090</PKey>
  </tkUnitDetail>
  <tkUnitDetail>
    <EndDate>2013-09-30T00:00:00-07:00</EndDate>
    <Employee>0A0001</Employee>
    <Seq>1</Seq>
    <TransDate>2013-09-17T00:00:00-07:00</TransDate>
    <WBS1>00BUDGET.001</WBS1>
    <WBS2 xml:space="preserve"/>
    <WBS3 xml:space="preserve"/>
    <UnitTable>9701300</UnitTable>
    <Unit>A0302</Unit>
    <Qty>15.0000</Qty>
    <PKey>TSAUDIT435861514091</PKey>
  </tkUnitDetail>
  <tkUnitDetail>
    <EndDate>2013-09-30T00:00:00-07:00</EndDate>
    <Employee>0A0001</Employee>
    <Seq>2</Seq>
```

```
      <TransDate>2013-09-18T00:00:00-07:00</TransDate>
      <WBS1>00BUDGET.002</WBS1>
      <WBS2>000N</WBS2>
      <WBS3 xml:space="preserve"/>
      <UnitTable>AA_timesheet</UnitTable>
      <Unit>A0103</Unit>
      <Qty>20.0000</Qty>
      <PKey>TSAUDIT435861514092</PKey>
    </tkUnitDetail>
    <tkUnitDetail>
      <EndDate>2013-09-30T00:00:00-07:00</EndDate>
      <Employee>0A0001</Employee>
      <Seq>2</Seq>
      <TransDate>2013-09-19T00:00:00-07:00</TransDate>
      <WBS1>00BUDGET.002</WBS1>
      <WBS2>000N</WBS2>
      <WBS3 xml:space="preserve"/>
      <UnitTable>AA_timesheet</UnitTable>
      <Unit>A0103</Unit>
      <Qty>10.0000</Qty>
      <PKey>TSAUDIT435861514093</PKey>
    </tkUnitDetail>
    <tkBreakTime>
      <EndDate>2013-09-30T00:00:00-07:00</EndDate>
      <Employee>0A0001</Employee>
      <TransDate>2013-09-16T00:00:00-07:00</TransDate>
      <BreaksTaken>2</BreaksTaken>
      <MealStartDateTime>2013-09-16T12:00:00-07:00</MealStartDateTime>
      <MealEndDateTime>2013-09-16T14:00:00-07:00</MealEndDateTime>
      <Meal2StartDateTime>2013-09-16T16:00:00-07:00</Meal2StartDateTime>
      <Meal2EndDateTime>2013-09-16T16:30:00-07:00</Meal2EndDateTime>
    </tkBreakTime>
    <tkBreakTime>
      <EndDate>2013-09-30T00:00:00-07:00</EndDate>
      <Employee>0A0001</Employee>
      <TransDate>2013-09-18T00:00:00-07:00</TransDate>
```

```
    <BreaksTaken>0</BreaksTaken>
    <MealStartDateTime>2013-09-18T12:00:00-07:00</MealStartDateTime>
    <MealEndDateTime>2013-09-18T13:00:00-07:00</MealEndDateTime>
  </tkBreakTime>
</ROOT>
```

Return Message

The Web method in the Web service is expected to return a string that is processed by Vision application code to either continue with the submit operation or abort it. If an empty string or empty error-XML is returned, it means that there were no errors or warnings during custom data validation. The timesheet is saved and marked as submitted. However, if any of the custom data validation rules fail, the Web method is expected to return error or warning messages as a string (in a specific format explained below). Those messages are displayed in Vision for the user submitting the timesheet. In such an event, the timesheet is not saved and the 'submit' transaction is not completed.

The return message is required to be in XML format to process and display it in Vision. The required format for the return XML message is as shown in the following example:

Error Message

```
<errors>
      <error>Error Message 1</error>
      <error>Error Message 2</error>
      <error>Error Message 3</error>
</errors>
```

Error messages 1, 2, and 3 are actual messages that will be displayed in Vision.

Warning Message

The returned message can also be designated as a warning rather than as an error. An XML attribute 'warning' is used to indicate that the message returned is to be displayed as a warning in Vision. The XML format for warning messages is exactly the same as the format used for error messages, but an extra attribute is used, as shown in the following example:

```
<errors warning="y">
      <error>Warning Message 1</error>
      <error>Warning Message 2</error>
      <error>Warning Message 3</error>
</errors>
```

Touch Time

If you use Vision Touch Time, the error or warning messages that you create cannot contain the pipe line (|) character. This character is reserved for messages that are returned from Vision Core timesheet validation to Touch Time.

Return Message to Indicate Successful Completion of Custom Data Validation

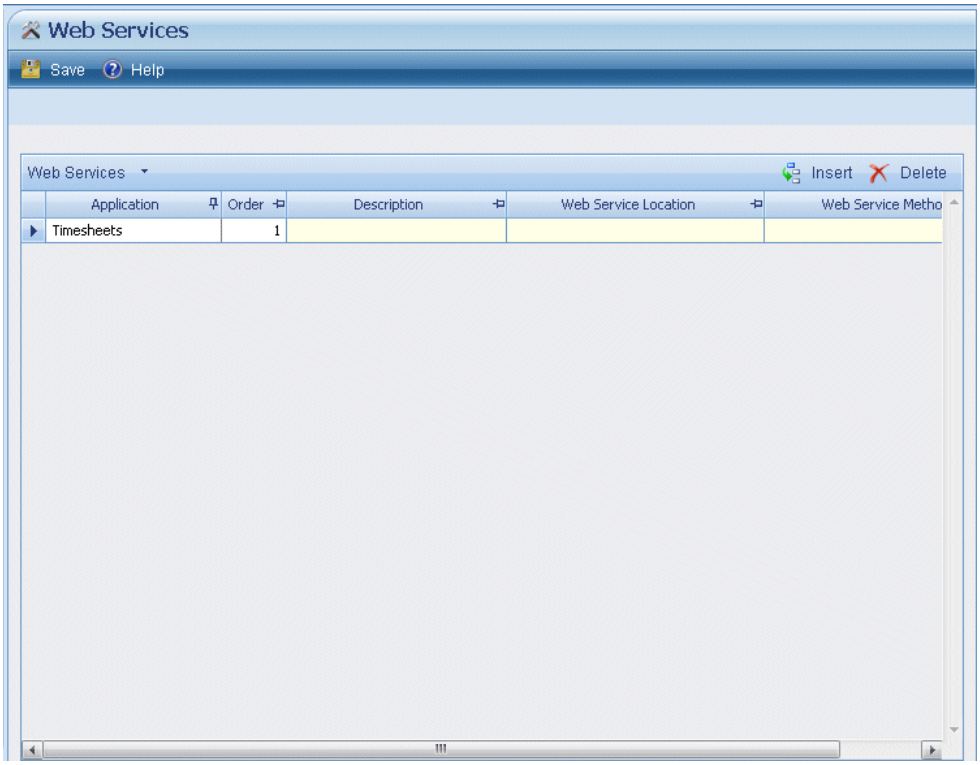
An empty string ('') or the following empty error XML message is returned.

```
<errors></errors>
```

Vision Timesheet Data Validation Options

Use the **Web Services** options to associate a Web service and a Web method with timesheet data validation during the 'submit' operation.

Vision: **Navigation Menu » Configuration » Workflow » Web Services**



Web Service Location — This can be either a location or file-path of the WSDL file generated for the Web service or it can be a URL to the WSDL located on a Web server. For Web services written in ASP.NET, it can also be a URL to the Web service asmx page with "?WSDL" added at the end. This parameter will generate the WSDL information automatically.

The following are some examples:

- WSDL File Location: \\Serv1\Share1\Proj1\VisionTSheetValidation.wsdl
- WSDL File URL: http://Serv1/VisionTSheetValService/VisionTSheetVal.wsdl
- ASMX page URL: http://Serv1/VisiontSheetValService/VisionTVal.asmx?wsdl
- Web Service Method — Based on the information that is contained in the WSDL file, a list of available Web methods in the Web service is loaded into the drop-down list. From the displayed list of Web methods, an appropriate method for expense report validation is selected.

Extending Data Validation Business Logic for Expense Reports

When expense reports are submitted in Vision, the existing data validation business logic can be extended to process custom validation rules using Web services. The custom validation process is treated as a part of single 'submit' transaction. In other words, based on the success or failure of the extended custom data validation, the entire submit operation is either completed or stopped.

The custom data validation process can return messages as errors or warnings, and the returned messages and errors are displayed in Vision. If no messages are returned, it is interpreted that custom data validation was successful, and the expense report is submitted and saved.

All of the custom data validation rules need to be encapsulated in one single Web service and one Web method is designated to be invoked when expense reports are submitted. In Vision, an expense report configuration option is available to specify the Web service and the Web method that need to be invoked when expense reports are submitted.

The Web service that is associated with the data validation of expense reports can also be used to launch other events and services that are outside and independent of Vision. At this time, it is not possible to invoke any internal workflow processes in Vision through the same Web service.

Implementation Details

When an expense report is submitted, after processing standard data validation rules, the specified Web service is called and the designated Web method in the Web service passes the parameter:

Parameter for the Web Method that Is Invoked when an Expense Report Is Submitted	
Expense report data/record (String – Entire expense report record in XML format)	This is the entire expense report record, which includes data from multiple expense-report-related tables in the Vision database, represented in XML format.

Expense Report Record in XML

As mentioned previously, the entire expense report record in XML format is passed to the Web method to process custom data validation business logic. A single expense report record consists mainly of data from two tables in the database:

- **ekMaster** — This table consists of a single row with master information for the record.
- **ekDetail** — This table is comprised of expense report details for each of the line items in the expense report. The custom data validation logic can be structured to validate any of the pieces of data in either the master or detail table. The XML format of the expense report record that is passed to the Web method is given below.

See the Vision Data Dictionary in the Vision online help for descriptions of the fields in tables.

```
- <ROOT>
- <ekMaster>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Status>N</Status>
  <AdvanceAmount>0.0000</AdvanceAmount>
  <Selected>N</Selected>
  <SubmittedBy>00001</SubmittedBy>
  <ApprovedBy>00001</ApprovedBy>
  <SubmittedDate>2009-03-25T00:00:00-07:00</SubmittedDate>
  <ApprovedDate>2008-11-03T00:00:00-08:00</ApprovedDate>
  <CreateUser>ADMIN</CreateUser>
  <CreateDate>2008-11-03T00:00:00-08:00</CreateDate>
  <ModUser>ADMIN</ModUser>
  <ModDate>2009-07-30T00:00:00-07:00</ModDate>
  <DefaultCurrencyCode xml:space="preserve" />
  <PaymentExchangeOverrideMethod>N</PaymentExchangeOverrideMethod>
  <PaymentExchangeOverrideRate>0.000000</PaymentExchangeOverrideRate>
  <CurrencyExchangeOverrideMethod>N</CurrencyExchangeOverrideMethod>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
    </ekMaster>
  <calledFrom>SaveClicked</calledFrom>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>1</Seq>
  <SortOrder>1</SortOrder>
  <TransDate>2008-11-03T00:00:00-08:00</TransDate>
  <Description>asdas</Description>
  <WBS1>1-9990-1500</WBS1>
  <WBS2 xml:space="preserve" />
```

```
<WBS3 xml:space="preserve" />
<Category>2</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.00</Account>
<Amount>1000.0000</Amount>
<Miles>2898.5500</Miles>
<AmountPerMile>0.3450</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>GST</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Code>HI</Tax2Code>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>2</Seq>
<SortOrder>2</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
```

```
<Account>521.01</Account>
<Amount>54.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>3.0600</TaxAmount>
<NetAmount>50.9400</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>54.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>3</Seq>
<SortOrder>3</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
```

```
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>4</Seq>
<SortOrder>4</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
```

```
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>5</Seq>
<SortOrder>5</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
```

```
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>6</Seq>
  <SortOrder>6</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>7</Seq>
```

```
<SortOrder>7</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>8</Seq>
<SortOrder>8</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
```

```
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>9</Seq>
<SortOrder>9</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
```

```
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>10</Seq>
<SortOrder>10</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
```

```
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>11</Seq>
<SortOrder>11</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
```

```
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>12</Seq>
  <SortOrder>12</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
```

```
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>13</Seq>
<SortOrder>13</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>14</Seq>
<SortOrder>14</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
```

```
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>15</Seq>
<SortOrder>15</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
```

```
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>16</Seq>
<SortOrder>16</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
```

```
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>17</Seq>
<SortOrder>17</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
```

```
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>18</Seq>
<SortOrder>18</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
```

```
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>19</Seq>
  <SortOrder>19</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
```

```
<Seq>20</Seq>
<SortOrder>20</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>21</Seq>
<SortOrder>21</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
```

```
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>22</Seq>
<SortOrder>22</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
```

```
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>23</Seq>
<SortOrder>23</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
```

```
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>24</Seq>
  <SortOrder>24</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
```

```
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>25</Seq>
  <SortOrder>25</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
```

```
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>26</Seq>
<SortOrder>26</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>27</Seq>
<SortOrder>27</SortOrder>
```

```
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>28</Seq>
<SortOrder>28</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
```

```
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>29</Seq>
<SortOrder>29</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
```

```
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>30</Seq>
<SortOrder>30</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
```

```
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>31</Seq>
<SortOrder>31</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
```

```
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>32</Seq>
  <SortOrder>32</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
```

```
<ReportName>aaa</ReportName>
<Seq>33</Seq>
<SortOrder>33</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>34</Seq>
<SortOrder>34</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
```

```
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>35</Seq>
<SortOrder>35</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
```

```
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>36</Seq>
<SortOrder>36</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
```

```
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>37</Seq>
<SortOrder>37</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
```

```
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>38</Seq>
  <SortOrder>38</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
```

```
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>39</Seq>
  <SortOrder>39</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>40</Seq>
```

```
<SortOrder>40</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>41</Seq>
<SortOrder>41</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
```

```
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>42</Seq>
  <SortOrder>42</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
```

```
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>43</Seq>
<SortOrder>43</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
```

```
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>44</Seq>
  <SortOrder>44</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
```

```
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>45</Seq>
  <SortOrder>45</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
```

```
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>46</Seq>
<SortOrder>46</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>47</Seq>
<SortOrder>47</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
```

```
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>48</Seq>
<SortOrder>48</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
```

```
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>49</Seq>
<SortOrder>49</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
```

```
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>50</Seq>
<SortOrder>50</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
```

```
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>51</Seq>
<SortOrder>51</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
```

```
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
  <Seq>52</Seq>
  <SortOrder>52</SortOrder>
  <TransDate>2009-01-09T00:00:00-08:00</TransDate>
  <WBS1>1-9990-0900</WBS1>
  <WBS2>002</WBS2>
  <WBS3>002</WBS3>
  <Category>3</Category>
  <Billable>Y</Billable>
  <CompanyPaid>N</CompanyPaid>
  <Account>521.01</Account>
  <Amount>1000.0000</Amount>
  <Miles>0.0000</Miles>
  <AmountPerMile>0.0000</AmountPerMile>
  <EditDetail>Y</EditDetail>
  <EKGroup>A</EKGroup>
  <TaxCode>CT</TaxCode>
  <TaxAmount>56.6000</TaxAmount>
  <NetAmount>943.4000</NetAmount>
  <PaymentExchangeRate>0.000000</PaymentExchangeRate>
  <PaymentAmount>1000.0000</PaymentAmount>
  <CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
  <Tax2Amount>0.0000</Tax2Amount>
  <CompoundTax>N</CompoundTax>
    </ekDetail>
- <ekDetail>
  <Employee>00001</Employee>
  <ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
  <ReportName>aaa</ReportName>
```

```
<Seq>53</Seq>
<SortOrder>53</SortOrder>
<TransDate>2009-01-09T00:00:00-08:00</TransDate>
<WBS1>1-9990-0900</WBS1>
<WBS2>002</WBS2>
<WBS3>002</WBS3>
<Category>3</Category>
<Billable>Y</Billable>
<CompanyPaid>N</CompanyPaid>
<Account>521.01</Account>
<Amount>1000.0000</Amount>
<Miles>0.0000</Miles>
<AmountPerMile>0.0000</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>CT</TaxCode>
<TaxAmount>56.6000</TaxAmount>
<NetAmount>943.4000</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>1000.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
- <ekDetail>
<Employee>00001</Employee>
<ReportDate>2008-11-03T00:00:00-08:00</ReportDate>
<ReportName>aaa</ReportName>
<Seq>54</Seq>
<SortOrder>54</SortOrder>
<TransDate>2009-07-14T00:00:00-07:00</TransDate>
<Category>1</Category>
<Billable>N</Billable>
```

```
<CompanyPaid>N</CompanyPaid>
<Account>103.00</Account>
<Amount>444.0000</Amount>
<Miles>1286.9600</Miles>
<AmountPerMile>0.3450</AmountPerMile>
<EditDetail>Y</EditDetail>
<EKGroup>A</EKGroup>
<TaxCode>MA</TaxCode>
<TaxAmount>21.1400</TaxAmount>
<NetAmount>422.8600</NetAmount>
<PaymentExchangeRate>0.000000</PaymentExchangeRate>
<PaymentAmount>444.0000</PaymentAmount>
<CurrencyExchangeOverrideRate>0.000000</CurrencyExchangeOverrideRate>
<Tax2Amount>0.0000</Tax2Amount>
<CompoundTax>N</CompoundTax>
  </ekDetail>
  </ROOT>
```

Return Message

The Web method in the Web service is expected to return a string that is processed by Vision application code to determine either to continue with the submit operation or to abort it. If an empty string or empty error-XML is returned, it means that there were no errors or warnings during custom data validation. The expense report is saved and marked as submitted. However, if any of the custom data validation rules fail, the Web method is expected to return error or warning messages as a string (in a specific format explained below). Those messages are displayed in Vision for the user who submitted the expense report. In such an event, the expense report is not saved and 'submit' transaction is not completed.

The return message is required to be in XML format to process and display it in Vision. The required format for the return XML message is shown in the following example:

Error Message

```
<errors>
      <error>Error Message 1</error>
      <error>Error Message 2</error>
      <error>Error Message 3</error>
</errors>
```

Error messages 1, 2, and 3 are actual messages that will be displayed in Vision.

Warning Message

A returned message can also be designated as a warning rather than as an error. An XML attribute ‘warning’ is used to indicate that the message returned is to be displayed as a warning in Vision. The XML format for warning messages is exactly the same as the format used for error messages, but an extra attribute is used, as shown in the following example:

```
<errors warning=”y”>
      <error>Warning Message 1</error>
      <error>Warning Message 2</error>
      <error>Warning Message 3</error>
</errors>
```

Return Message to Indicate Successful Completion of Custom Data Validation

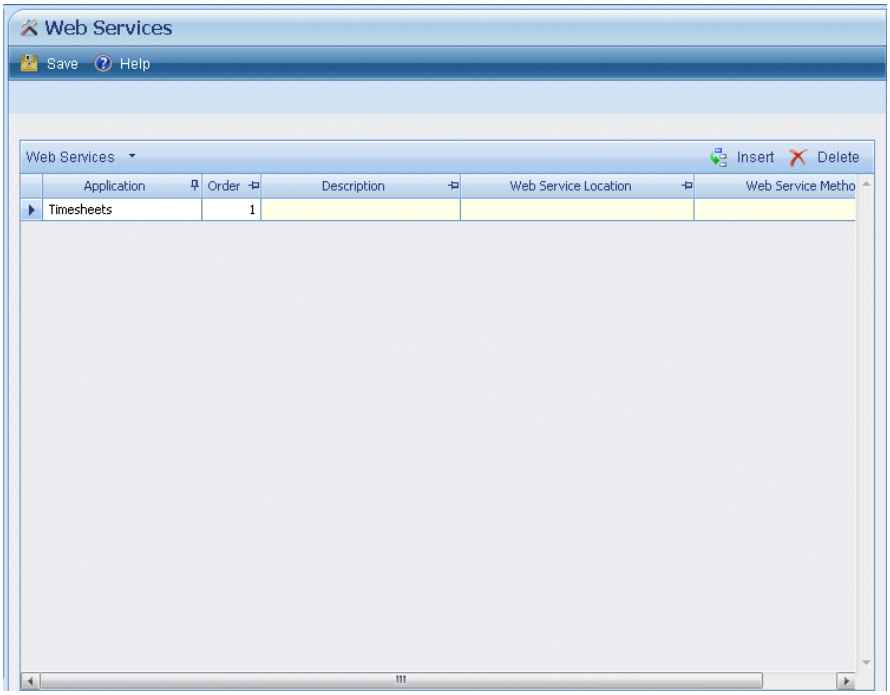
An empty string ('') or the following empty error XML message is returned.

```
<errors></errors>
```

Vision Expense Report Data Validation Options

Use the **Web Services** options to associate a Web service and a Web method with expense report data validation during the ‘submit’ operation.

Vision: **Navigation menu » Configuration » Workflow » Web Services**



- **Web Service Location —** This can be either a location or file path of the WSDL file generated for the Web service, or it can be a URL to the WSDL located on a Web server. For Web services written in ASP.NET, it can also be a URL to the Web service asmx page with “?WSDL” added at the end. This parameter will generate the WSDL information automatically. The following are some examples:

- WSDL File Location: \\Serv1\Share1\Proj1\VisionExpenseValidation.wsdl
- WSDL File URL: http://Serv1/VisionExpenseValService/VisionExpenseVal.wsdl
- ASMX page URL: http://Serv1/VisionExpenseValService/VisionEVal.asmx?wsdl

- **Web Service Method —** Based on the information contained in the WSDL file, a list of available Web methods in the Web service prefills in the drop-down list. Select an appropriate Web method for expense report validation from the list.

When Web service location is provided, Vision reads the WSDL information to gather the list of Web methods. In the event of any problems with either locating the WSDL file or reading from it, an exception is captured and an error message is displayed.

Enable Workflows for APIs

Workflows can be triggered in Vision when Info Center records are added or updated in Vision with VisionXtend APIs. To enable workflows to be triggered, you must add the WebAPI.EnableWorkflow setting to the **<appSettings>** section of the web.config file on the Vision web server.

Enter the following in the <appSetting> section of the web.config file:

```
<add key="WebAPI.EnableWorkflow" value="Y" />
```

A “Y” value triggers the workflows. Workflows are not triggered if the setting has an "N" value or the setting does not exist in the web.config file.

Connect for Microsoft API Calls

Connect for Microsoft API calls are not affected by the WebAPI.EnableWorkflow setting. Workflows are always triggered by the Connect for Microsoft Outlook API calls.

Add or Change the WebAPI.EnableWorkflow Setting

To add or change the WebAPI.EnableWorkflow setting, complete the following steps:

1. Back up the existing web.config file on the Vision web server.

   The default location is C:\Program Files\Deltek\Vision\Web.

2. Open Windows Notepad using the **Run as administrator** option.

   To do this, right-click the Notepad program icon, and on the shortcut menu, click **Run as administrator**.

3. In Notepad, click **File » Open**.

4. In the Open dialog box, navigate to the Vision installation folder (specified in step 1), select the web.config file, and click **Open**.

5. In the web.config file displayed in Notepad, navigate to the <appSettings> section, and add the WebAPI.EnableWorkflow setting below all the other <add key=..... /> line entries if it does not already exist:

   ```
   <add key="WebAPI.EnableWorkflow" value="Y" />
   ```

   “Y” triggers the workflows, “N” or anything else does not.

6. Save and close the file.

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