Deltek VisionXtend™

Invoking a Custom Method to Process Workflow Actions

Deltek Vision 6.1

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

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

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

This edition published June, 2009

Contents

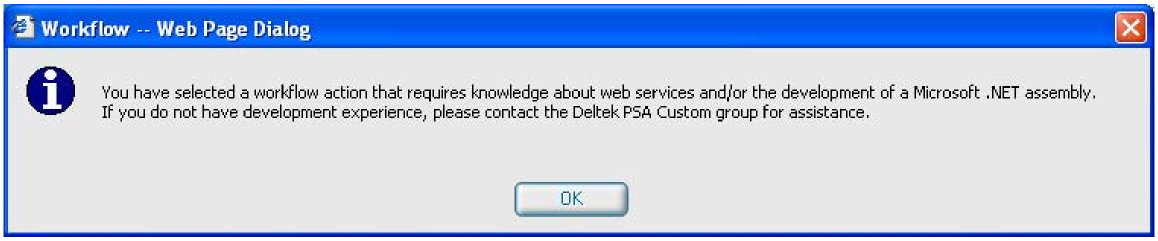
Overview

The Workflow feature in Vision is very comprehensive and 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 such option 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 document. The code excerpts used for examples below are in VB.Net.

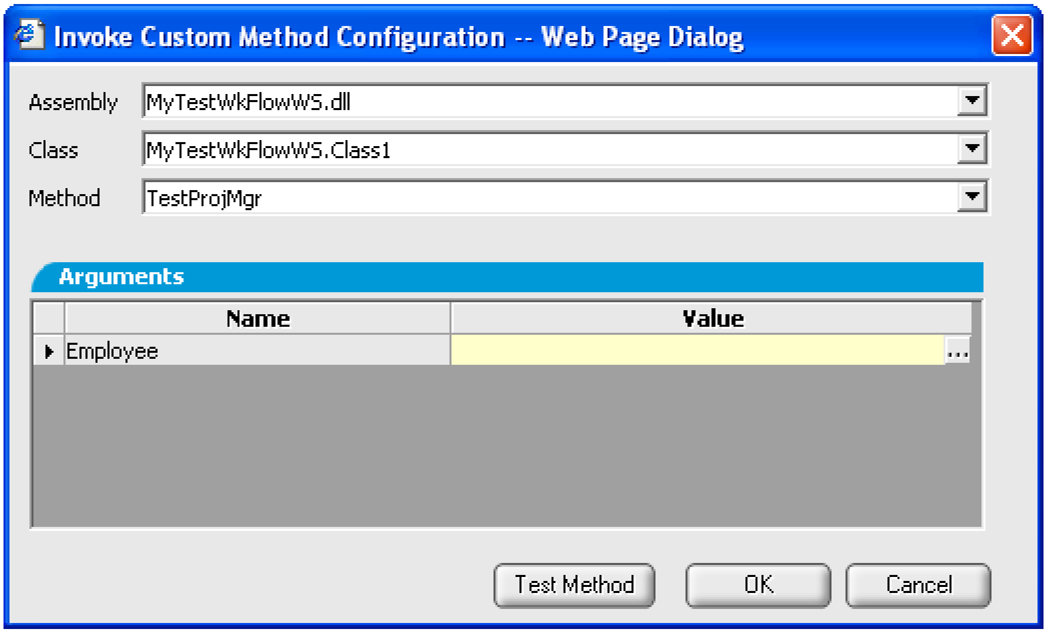
Implementation Details

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 this option is selected, Vision displays the following warning message.



You have selected a workflow action that requires knowledge about web services and/or the development of a Microsoft .NET assembly.
If you do not have development experience, please contact the Deltek PSA Custom group for assistance.

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



Assembly: MyTestWkFlowWS.dll

Class: MyTestWkFlowWS.Class1

Method: TestProjMgr

Name	Value
Employee	

Fields on the Invoke Custom Method Configuration Dialog

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

Field	Description
Assembly	The name of the actual Microsoft.Net assembly or the DLL that consists of the custom method that is being invoked. The dropdown list is automatically populated based on Microsoft.Net assemblies or DLLs that are present in <*VisionInstallDir*>*\Workflow* folder. Note that assemblies or DLLs that are used for invoking custom methods are required to be placed in the above mentioned directory. Vision does not provide the option of using an assembly that is located elsewhere.

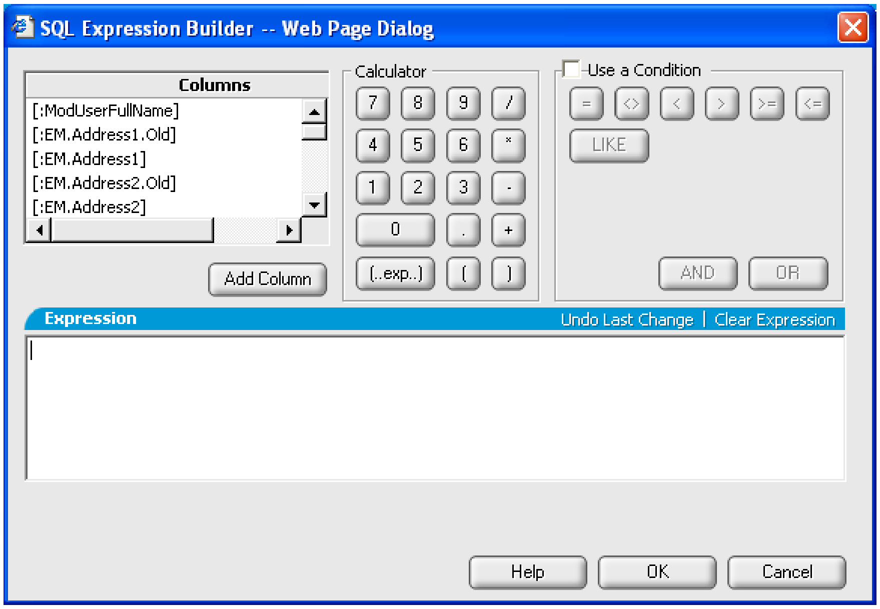
Field	Description
Class	When a specific assembly is selected from the list, Vision automatically reads the assembly and loads the **Class** dropdown with a list of classes available in the assembly.
Method	The **Method** dropdown field is automatically loaded with a list of methods (functions or subs) available in the selected class. The selected method from the list is the method that will be invoked when workflow conditions are met.
Arguments	The Arguments grid is automatically loaded with a list of arguments or parameters defined for the selected method. 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 users to build simple or complex SQL expressions using Vision database fields. The expression can be as simple as just 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 the **Add Column** button is used. All numeric type fields are added with no qualifiers.

The SQL expressions are not validated or evaluated when they are being built. Deltek recommends checking expressions for accuracy before saving them.



Testing the Method

Once all of the configuration information has been filled in, the **Test Method** button on the Invoke Custom Method Configuration dialog can be used to test if 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 below 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.

Once 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	Add an error message to display in Vision. 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	Add an error message to display as fatal in Vision. 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	Add a message to display as information in Vision. **Parameter**: Information Message as String **Returns**: N/A
AddWarning	Add a message to display as warning in Vision. **Parameter**: Warning Message as String **Returns**: N/A

Name	Description
ExecuteSQL	Execute a SQL query on the database. 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	Execute a SQL query on the database and return back data. 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 hence eliminates the possibility of deadlocks in the database. **Parameter**: SQL Query as String **Returns**: DataTable (VB.Net)

WorkflowAPIMethods

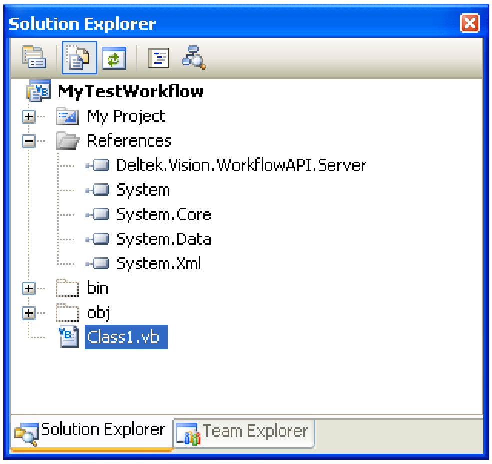
The WorkflowAPIMethods class consists of the following functions that can be called from a custom method. For details about parameters and return values, refer to WorkflowCustomHelp.chm help file.

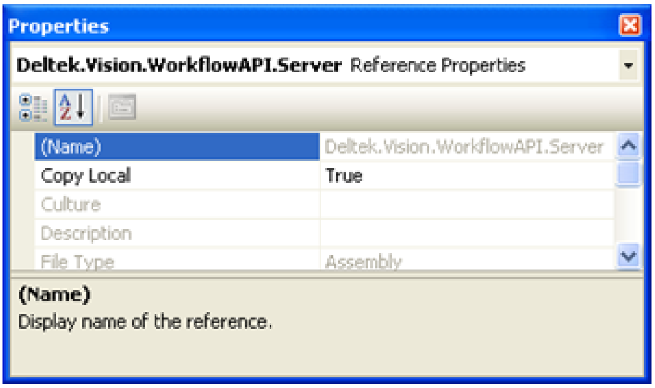
The naming convention used for Info Centers below is based on the standard names used in Vision. These Info Center names may be different in cases where Info Center names have been changed.

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

Illustrated Examples with Code

When Deltek.Vision.WorkflowAPI.Server.dll 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, as shown below. This will stop the copying of the referenced DLL to the output directory and hence it does not complain about the dependencies of Deltek.Vision.WorkflowAPI.Server.dll when compiling your custom assembly.





Deltek.Vision.WorkflowAPI.Server Reference Properties

(Name)	Deltek.Vision.WorkflowAPI.Server
Copy Local	True
Culture	
Description	
File Type	Assembly

(Name)
Display name of the reference.

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 example above, 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 below and any applicable function from the class is called.

```
Dim bReturnValue As Boolean

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

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