



**Q-Pulse Web Service APIs
Dynamically Linking a Word Document to
Metadata of a Q-Pulse Document**

Contents

Introduction.....	3
Assumptions	3
Overview of the Example	3
Strategy	3
Creating a Word Template	3
Setting up Q-Pulse Document Retrieval Support	4
Reading a Word Document Form Field Value.....	4
Reading a Word Document Property	5
Updating a Word Document Form Field	6
Updating a Word Document Property.....	7
Implementing Q-Pulse Web Service Authentication.....	8
Implementing Retrieval of Metadata of a Q-Pulse Document by Document Number and Revision Reference	10
Mapping Metadata of a Q-Pulse Document to Form Fields and Properties	11
Implementing an Event Handler to Receive Word Events	14
Creating a Word Document Instance using the template	15
Limitations/Improvements	16

Introduction

The purpose of this tutorial is to demonstrate the possibility of dynamically linking the metadata of a Q-Pulse document retrieved using the Q-Pulse Web Services into the fields and properties of a Word Document.

Assumptions

The user is assumed to be familiar with VBA Word development. The user is using Microsoft Word 2003 or later. Note: The instructions are written for Word 2003 and so later versions may have different menu paths/shortcuts/etc to access certain features but ultimately the approach involved is the same. Macro support in Word is enabled or configured to ask user to enable on opening of document.

Overview of the Example

The example application should initially allow the user to create a Word Document Template which allows the retrieval of metadata of a particular Q-Pulse document revision. It will then show how to map the retrieved metadata to Word document fields and properties of the template and document instance. Finally it will show how to create an instance of a Word document based on the template which links to a specific revision of a Q-Pulse document.

Strategy

Word Documents allow modification of document elements through the use of VBA macro code. VBA macro code also has support for making HTTP requests which provides the opportunity to use the Q-Pulse Web Services APIs to retrieve Q-Pulse Document metadata. A combination of these features allows the dynamic mapping of the metadata to Word document fields and properties. Creating a Word Template with this functionality allows users to create Word document instances with this support. The majority of the functionality will be contained in the VBA Modules within the Word Template.

Creating a Word Template

1. Open Microsoft Word and create a new blank document by going to File -> New Project -> Blank Document.
2. Save blank document as a template by going to File -> Save As and selecting Save As Type 'Document Template (*.dot)'. Choose an appropriate name to save template e.g. QPulseWordDocument.dot

Setting up Q-Pulse Document Retrieval Support

1. In the word template open up the Visual Basic Editor by going to Tools -> Macro -> Visual Basic Editor or pressing Alt+F11
2. In Project Explorer, expand the Project that contains the name given to the template in brackets e.g. Project (QPulseWordDocument)
3. Right click on the folder named 'Modules' and select Insert -> Module
4. Rename the module to a suitable name e.g. modQPulseWebAPI by editing its Name property (If Properties dialog is not visible open it by selecting View -> Properties Window)

Reading a Word Document Form Field Value

1. In the modQPulseWebAPI module add a Function to find and read the value of a form field from a list of form fields

```
Function GetFieldObjValueInList (name As String,
                                ByVal fieldList As Fields)
On Error GoTo ErrHandlerGetFieldObjValueInList
Dim f As Word.Field
Dim tb As TextBox
Dim lb As Label
For Each f In fieldList
If f.OLEFormat Is Nothing Then ' do nothing
Else
    If f.OLEFormat.ClassType = "Forms.TextBox.1" Then ' it
is a textbox
        Set tb = f.OLEFormat.Object
        If StrComp(tb.name, name, vbTextCompare) = 0 Then
            GetFieldObjValueInList = tb.Text
            Exit For
        End If
    ElseIf f.OLEFormat.ClassType = "Forms.Label.1" Then '
it is a label
        Set lb = f.OLEFormat.Object
        If StrComp(lb.name, name, vbTextCompare) = 0 Then
            GetFieldObjValueInList = lb.Caption
            Exit For
        End If
    End If
End If ' f.oleformat is nothing
Next
fErrHandlerGetFieldObjValueInList:
    Err.Clear
    GetFieldObjValueInList = ""
End Function
```

2. In the modQPulseWebAPI module add a Function to find and read the value of a form field in a Word Document Object Model

```
Function GetFieldObjValue(myActiveDocument As Document,
                        name As String)
On Error GoTo ErrHandlerGetFieldObjValue
Dim result
' Check main doc fields
result = GetFieldObjValueInList(name,
myActiveDocument.Fields)
If Not IsEmpty(result) And Not result = vbNullString Then
    GetFieldObjValue = result
    Exit Function
End If
' Check doc header fields
result = GetFieldObjValueInList(name,
myActiveDocument.Sections(1).Headers(wdHeaderFooterPrimary).
Range.Fields)
If Not IsEmpty(result) And Not result = vbNullString Then
    GetFieldObjValue = result
    Exit Function
End If
' Check doc footer fields
result = GetFieldObjValueInList(name,
myActiveDocument.Sections(1).Footers(wdHeaderFooterPrimary).
Range.Fields)
GetFieldObjValue = result
Exit Function
ErrHandlerGetFieldObjValue:
    Err.Clear
    GetFieldObjValue = ""
End Function
```

Reading a Word Document Property

3. In the modQPulseWebAPI module add a Function to find and read the value of a Word document property

```
Function ReadProperty(myActiveDocument As Document,
                    name As String)

Dim custom As Boolean
Dim value As String
On Error GoTo ErrHandlerReadProperty
' Try built-in properties first
```

```

value =
myActiveDocument.BuiltInDocumentProperties(name).value
ReadProperty = value
Exit Function
' Try custom properties second
ContinueCustom:
    custom = True
GetCustomProperty:
    value =
myActiveDocument.CustomDocumentProperties(name).value
    ReadProperty = value
    Exit Function
ErrorHandlerReadProperty:
    Err.Clear
    If Not custom Then
        Resume ContinueCustom
    Else
        ReadProperty = ""
        Exit Function
    End If
End Function

```

Updating a Word Document Form Field

1. In the modQPulseWebAPI module add a Function to find and update the value of a form field from a list of form fields

```

Function SetFieldObjValueInList(name As String,
                                value As String,
                                ByVal fieldList As Fields)

Dim f As Word.Field
Dim tb As TextBox
Dim lb As Label
For Each f In fieldList
If f.OLEFormat Is Nothing Then ' do nothing
Else
    If f.OLEFormat.ClassType = "Forms.TextBox.1" Then ' it
is a textbox
        Set tb = f.OLEFormat.Object
        If StrComp(tb.name, name, vbTextCompare) = 0 Then
            tb.Text = value
            Exit For
        End If
    ElseIf f.OLEFormat.ClassType = "Forms.Label.1" Then '
it is a label
        Set lb = f.OLEFormat.Object
    End If
End For

```

```

        If StrComp(lb.name, name, vbTextCompare) = 0 Then
            lb.Caption = value
            Exit For
        End If
    End If
End If ' f.oleformat is nothing
Next
End Function

```

2. In the modQPulseWebAPI module add a Function to find and update the value of a form field in a Word Document Object Model

```

Function GetFieldObjValue(myActiveDocument As Document,
                        name As String)

Dim result
' Check main doc fields
result = SetFieldObjValueInList(name, value,
myActiveDocument.Fields)
' Check doc header fields
result = SetFieldObjValueInList(name, value,
myActiveDocument.Sections(1).Headers(wdHeaderFooterPrimary).
Range.Fields)
' Check doc footer fields
result = SetFieldObjValueInList(name, value,
myActiveDocument.Sections(1).Footers(wdHeaderFooterPrimary).
Range.Fields)
End Function

```

Updating a Word Document Property

1. In the modQPulseWebAPI module add a Function to find and update (or add if doesn't exist) the value of a Word document property

```

Function WriteProperty(myActiveDocument As Document,
                    name As String,
                    value As String)

Dim custom As Boolean
On Error GoTo ErrHandlerWriteProperty
myActiveDocument.BuiltInDocumentProperties(name).value =
value
Exit Function
ContinueCustom:
    custom = True
SetCustomProperty:
myActiveDocument.CustomDocumentProperties(name).value =
value
Exit Function

```

```

AddCustomProperty:
    On Error Resume Next
    myActiveDocument.CustomDocumentProperties.Add
name:=name, LinkToContent:=False, Type:=
msoPropertyTypeString, value:=value
    Exit Function
ErrorHandlerWriteProperty:
    Select Case Err
        Case Else
            Err.Clear
            If Not custom Then
                Resume ContinueCustom
            Else
                Resume AddCustomProperty
            End If
        End Select
    End Function

```

Implementing Q-Pulse Web Service Authentication

1. In the modQPulseWebAPI module add a Function to perform a login to the Q-Pulse Web Core Service and return an Authentication token for the current session

```

Function PerformLogin(ByVal username As String,
                    ByVal password As String,
                    ByVal databasename As String)
    ' Create HTTP Request Object
    ...
    ' Build up XML string of Q-Pulse Authenticate SOAP Data
    Contract
    ...
    ' Perform HTTP SOAP Action On Q-Pulse Core Service
    ...
    ' Extract authentication token from XML SOAP response string
    ...
End Function

```

2. Insert code to create a HTTP Request object

```

Dim httpRequest
' Create HTTP Request Object
Set httpRequest = CreateObject("Msxml2.XMLHTTP")

```

3. Insert code to build up XML string of Q-Pulse Authenticate SOAP Data Contract

```

Dim loginRequest
' Build up XML string of Q-Pulse Authenticate SOAP Data
Contract
loginRequest = "<s:Envelope
xmlns:s=""http://schemas.xmlsoap.org/soap/envelope/"">"
loginRequest = loginRequest + "<s:Body>"
loginRequest = loginRequest + "<Authenticate
xmlns=""http://www.qpulse.com/QPulseWebServices/v1.1/"">"
loginRequest = loginRequest + "<userName>" + username +
"</userName>"
loginRequest = loginRequest + "<password>" + password +
"</password>"
loginRequest = loginRequest + "<database>" + databasename +
"</database>"
loginRequest = loginRequest + "</Authenticate>"
loginRequest = loginRequest + "</s:Body>"
loginRequest = loginRequest + "</s:Envelope>"

```

4. Insert code to perform HTTP SOAP Action on Q-Pulse Core Service changing the Q-Pulse Web Services URL suitable to your setup

```

' Perform HTTP SOAP Action On Q-Pulse Core Service
httpRequest.Open "POST",
"http://localhost/QPulse5WebServices/Services/Core.svc",
False
httpRequest.setRequestHeader "Content-Type", "text/xml"
httpRequest.setRequestHeader "SOAPAction",
http://www.qpulse.com/QPulseWebServices/v1.1/ICore/Authenticate
httpRequest.send loginRequest

```

5. Insert code to extract and return the Authentication token from the XML SOAP response string

```

' Extract authentication token from XML SOAP response string
Set objXMLDoc = CreateObject("Microsoft.XMLDOM")
objXMLDoc.async = False
objXMLDoc.loadXML(httpRequest.responseText)
Set node = objXMLDoc.documentElement
Set NodeList =
node.getElementsByTagName("AuthenticateResult")
For Each Elem In NodeList
    token = Elem.FirstChild.NodeValue
Next
' Return token value
PerformLogin = token

```

Implementing Retrieval of Metadata of a Q-Pulse Document by Document Number and Revision Reference

1. In the modQPulseWebAPI module add a Function to perform the retrieval of metadata of a particular revision of a Q-Pulse document by supplying its document Number and Revision Reference

```
Function GetDocByNumAndRevisionRef(ByVal token As String,
                                   ByVal documentNumber As String,
                                   ByVal documentRevisionReference As String)
    ' Build up XML string of Q-Pulse
    GetDocumentByRevisionReference SOAP Data Contract
    ...
    ' Perform HTTP SOAP Action On Q-Pulse Document Service
    ...
    ' Return XML SOAP response string
    ...
End Function
```

2. Insert code to create a HTTP Request object

```
Dim httpRequest
' Create HTTP Request Object
Set httpRequest = CreateObject("Msxml2.XMLHTTP")
```

3. Insert code to build up XML string of Q-Pulse Login SOAP Data Contract

```
Dim docRequest
' Build up XML string of Q-Pulse
GetDocumentByRevisionReference SOAP Data Contract
loginRequest = "<s:Envelope
xmlns:s=""http://schemas.xmlsoap.org/soap/envelope/"">"
docRequest = docRequest + "<s:Body>"
docRequest = docRequest + "<GetDocumentByRevisionReference
xmlns=""http://www.qpulse.com/QPulseWebServices/v1.1/"">"
docRequest = docRequest + "<token>" + token + "</token>"
docRequest = docRequest + "<documentNumber>" +
documentNumber + "</documentNumber>"
docRequest = docRequest + "<documentRevisionReference >" +
documentRevisionReference + "</documentRevisionReference>"
docRequest = docRequest +
"</GetDocumentByRevisionReference>"
docRequest = docRequest + "</s:Body>"
docRequest = docRequest + "</s:Envelope>"
```

4. Insert code to perform HTTP SOAP Action on Q-Pulse Core Service changing the Q-Pulse Web Services URL suitable to your setup

```
' Perform HTTP SOAP Action On Q-Pulse Document Service
httpRequest.Open "POST",
"http://localhost/QPulse5WebServices/Services/Document.svc",
False
httpRequest.setRequestHeader "Content-Type", "text/xml"
httpRequest.setRequestHeader "SOAPAction",
http://www.qpulse.com/QPulseWebServices/v1.1/IDocument/GetDocumentByRevisionReference
httpRequest.send docRequest
```

5. Return XML SOAP Response string

```
GetDocByNumAndRevisionRef = httpRequest.reponseText
```

Mapping Metadata of a Q-Pulse Document to Form Fields and Properties

1. In the modQPulseWebAPI module add a Function to find and read the value of a form field or property in a Word document

```
Function GetFieldAndPropertyValue (
    myActiveDocument As Document,
    name As String)

Dim result
result = ReadProperty(myActiveDocument,name)
If IsNull(result) Or IsEmpty(result) Or result =
vbNullString Then
    result = GetFieldObjValue(myActiveDocument,name)
End If
GetFieldAndPropertyValue = result
End Function
```

2. In the modQPulseWebAPI module add a Function to find and update the value of form fields or properties in a Word document

```
Function SetFieldAndPropertyValue (
    myActiveDocument As Document,
    name As String,
    value As String)

Dim result
result = WriteProperty(myActiveDocument,name,value)
result = GetFieldObjValue(myActiveDocument,name,value)
SetFieldAndPropertyValue = result
```

End Function

3. In the modQPulseWebAPI module add a Function to map the metadata in a Document SOAP response to each form field or property with a matching name

```
Function UpdateDocumentByRevisionReference (
    myActiveDocument As Document,
    ByVal docSoapReponse As String)

Dim updateResult
Dim resultNodePath
Dim childNodePath
Set objXMLDoc = CreateObject("Microsoft.XMLDOM")
objXMLDoc.async = False
objXMLDoc.loadXML (docSoapReponse)
' Extract Document Details
resultNodePath =
    "//s:Envelope/s:Body/GetDocumentByRevisionReferenceResponse/
    GetDocumentByRevisionReferenceResult"
Set resultNode =
objXMLDoc.documentElement.SelectSingleNode(resultNodePath)
If IsNull(resultNode) Or resultNode Is Nothing Then
    UpdateDocumentByRevisionReference = "Not Found"
    Exit Function
End If
Dim Number
Number = " "
Set NodeList = resultNode.ChildNodes
For Each Elem In NodeList
    If Elem.NodeTypeString = "element" Then
        If Not Elem.HasChildNodes Or Elem.ChildNodes.Length
= 0 Then
            ' Do nothing
        ElseIf Elem.HasChildNodes And Elem.ChildNodes.Length
= 1 Then
            'Text Elements have a child which is the text
            updateResult =
SetFieldAndPropertyValue(myActiveDocument, Elem.BaseName,
Elem.Text)
        Else
            childNodePath =
            "//s:Envelope/s:Body/GetDocumentByRevisionReferenceResponse/
            GetDocumentByRevisionReferenceResult/" + Elem.BaseName +
            "/IsSet"
            Set childNode =
Elem.SelectSingleNode(childNodePath)
```

```

        If IsNull(childNode) Or childNode Is Nothing
Then
            Set lastChildNode = Elem.LastChild
            If Not IsNull(lastChildNode) And Not
lastChildNode Is Nothing Then
                updateResult =
SetFieldAndPropertyValue(myActiveDocument, Elem.BaseName,
lastChildNode.Text)
            End If
        Else
            If childNode.Text = "true" Then
                Set valueChildNode =
Elem.SelectSingleNode("Value")
                updateResult =
SetFieldAndPropertyValue(myActiveDocument, Elem.BaseName,
valueChildNode.Text)
            End If
        End If
    End If
End If
Next
UpdateDocumentByRevisionReference = updateResult
End Function

```

4. In the modQPulseWebAPI module add a Function to control the retrieval of metadata and the mapping of it to a document

```

Function UpdateDocByRevisionRef(myActiveDocument As
Document)
Dim docSoapReponse
Dim docNum
Dim docRev
Dim updateResult
Dim token
' Confirm are logged in and have authentication token
If IsNull(token) Or IsEmpty(token) Or token = vbNullString
Then
    token = PerformLogin()
End If
docNum = GetFieldAndPropertyValue(myActiveDocument, "Number")
docRevRef =
GetFieldAndPropertyValue(myActiveDocument, "RevisionReference
")
If Not IsNull(docNum) And Not IsEmpty(docNum) And
Not docNum = vbNullString And Not IsNull(docRevRef) And
Not IsEmpty(docRevRef) And Not docRevRef = vbNullString
Then
    docSoapReponse = GetDocumentByRevisionReference(token,

```

```

                                                docNum, docRevRef)
        UpdateDocByRevisionRef =
        UpdateDocumentByRevisionReference(myActiveDocument,
                                                docSoapReponse)
    EndIf
End Function

```

Implementing an Event Handler to Receive Word Events

1. In Project Explorer, expand the Project that contains the name given to the template in brackets e.g. Project (QPulseWordDocument)
2. Right click on the folder named 'Class Modules' and select Insert -> Class Module
3. Rename the class module to a suitable name e.g. clsEventHandler by editing its Name property (If Properties dialog is not visible open it by selecting View -> Properties Window)
4. In the clsEventHandler class module add a declaration for an event handler

```

Public WithEvents QPulseWordDocEventHandler As
Word.Application

```

5. In the clsEventHandler class module add a Sub procedure to be run on a DocumentOpen event

```

Private Sub QPulseWordDocEventHandler_DocumentOpen(ByVal Doc
As Document)
Dim updateResult
updateResult = UpdateDocByRevisionRef(Doc)
End Sub

```

6. In the clsEventHandler class module add a Sub procedure to be run on a DocumentNew event

```

Private Sub QPulseWordDocEventHandler_NewDocument(ByVal Doc
As Document)
Dim updateResult
updateResult = UpdateDocByRevisionRef(Doc)
End Sub

```

7. In Project Explorer, expand the Project that contains the name given to the template in brackets e.g. Project (QPulseWordDocument)
8. Right click on the folder named 'Modules' and select Insert -> Module
9. Rename the module to a suitable name e.g. modHandleEvents by editing its Name property (If Properties dialog is not visible open it by selecting View -> Properties Window)
10. In the modHandleEvents module declare a variable and add a Sub procedure to create an instance of the clsEventHandler class module

```

Dim objEventHandler As clsEventHandler

```

```

Sub CreateEventHandler()
    'create a new event handler object and assign it to
objEventHandler
    Set objEventHandler = New clsEventHandler
    'Notify the current instance of Word to check for
    'application event procedures
    'inside the event handler object
    Set objEventHandler.QPulseWordDocumentEventHandler =
Word.Application
End Sub

```

11. In the modHandleEvents module implement the AutoExec template method to automatically create the event handle when the template method is executed

```

Sub AutoExec()
    CreateEventHandler
End Sub

```

12. In the modHandleEvents module implement the AutoNew template method to automatically create the event handle when the template method is executed

```

Sub AutoNew()
    CreateEventHandler
End Sub

```

13. In the modHandleEvents module implement the AutoOpen template method to automatically create the event handle when the template method is executed

```

Sub AutoOpen()
    CreateEventHandler
End Sub

```

Creating a Word Document Instance using the template

1. Save and Close Visual Basic Editor and Word Template
2. Create a document using the template
3. Insert into the new blank document a label form field with the name 'Number'
4. Insert into the new blank document a label form field with the name 'RevisionReference'
5. Insert into the new blank document a label form field with the name 'Status'
6. Insert into the new blank document a label form field with the name 'DocumentID'
7. Save the new document with an appropriate name e.g. QPulseSampleDoc and close it
8. Right Click on the file and select Properties and navigate to Custom tab.
9. Add a custom property with the name 'Number' and add a value of the required Q-Pulse document number to retrieve
10. Add a custom property with the name 'RevisionReference' and add a value of the required Q-Pulse document revision reference to retrieve
11. Click Ok

12. Open up word document and see that the form fields will have been updated to reflect the metadata of the document revision retrieved

Limitations/Improvements

As this is an example, in a real world deployment scenario there would be other considerations to address. Here are a few suggestions:

- The Web Service calls would be made across an encrypted connection (SSL).
- Any exception or error occurring while communicating with the Web Service would be handled and the user notified accordingly.
- Location of Form Field controls in the Word Document Object Model (DOM) will change when formatting is applied to them and so the Function which searches the DOM for form fields may need modified to look in the appropriate DOM locations
- To avoid having to enter both a Document Number and Revision Reference it is possible to use the Q-Pulse Document Web Service 'GetDocumentByRevisionID' to retrieve a specific document revision.



Gael Ltd.

Orion House,
S. E. Technology Park,
East Kilbride,
Scotland. G75 0RD

t: +44 1355 593400

f: +44 1355 579191

e: info@gaelquality.com

w: www.gaelquality.com

Q-Pulse is a registered trademark of Gael Products Ltd. All rights reserved worldwide.
Copyright © 2010 Gael Products Ltd. Gael Quality is a trading division of Gael Ltd.