

Creating Sales Receipts In QuickBooks With Access VBA:

A Feasibility Study and Demonstration in the Application of the QuickBooks SDK

Presented to:
Bill Skowera &
Kevin Hawkins

For Cedarstone Partners

By Scott B. Bradley
June 8, 2004

Table of Contents

Preliminary Notes	1
Introduction.....	1
Application Demonstration.....	2
The Code.....	4
Conclusion	5
Appendix A – VBA Code.....	A-1

Preliminary Notes

This report documents the Access VBA project for the demonstration of the feasibility of generating Sales Receipts in QuickBooks from within Access. The following files are required for the demonstration:

Microsoft Access File:

I:\Access Databases\mdb Files\TestDB_SBB.mdb

QuickBooks Enterprise File:

I:\Quickbooks Enterprise Companies\TestDataOnly_SBB.QBW

With these files open, the procedures in this document may be followed.

Introduction

Toward the end of developing a more streamlined and efficient process of check processing at Cedarstone, this feasibility study has been requested in order to assess the potential for automating Sales Receipt Transactions in client QuickBooks files. This is part of an ongoing project of re-engineering the data-entry and receipting processes. The role of this study in the overall redevelopment plan is shown in Figure 1.

Check Scanner Creates CSV	Not demonstrated in this study.
CSV is parsed for Receipt Data	
Parsed data is formatted as QBXML	Reasonably demonstrated in this study.
QBXML is passed to Quickbooks	
Sales Receipt is generated	

Figure 1. Scope of This Study

This study makes its case through the demonstration of Sales Receipts creation through a form interface in Access, and does not demonstrate the following:

- Automatic parsing of grouped data (such as CSV files from a check scanner or spreadsheet)
- Matching of donors with expected designation (as might be accomplished with a table of standard donations)

The form demonstration allows for any combination of donor/designation match and transaction information (date, type, check number, amount) and so may result in unrealistic data – yet adequately demonstrates the feasibility of generating Sales Receipts as required.

Application Demonstration

The Create Sales Receipt Form from the Access application is shown in Figure 2. This form can be used to create a Sales Receipt in QuickBooks by specification of required input parameters. All parameters are required except Check Number, which is optional.

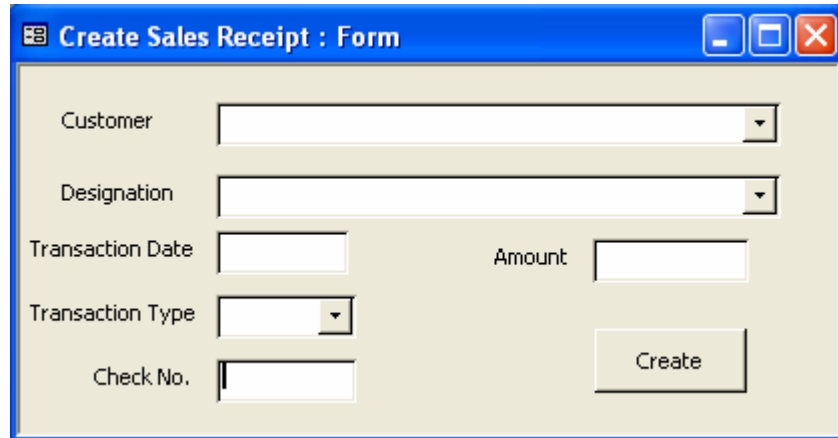


Figure 2. Sales Receipt Creation Form

To create a sales receipt select and fill-in the form as appropriate, and click Create. If there is an attempt to create a receipt without all the needed information, Access will provide a notification similar to that shown in Figure 3.

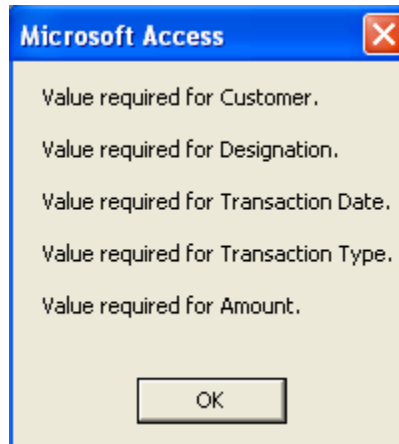


Figure 3. Null Value Message Box

After filling in the form, it might look something like Figure 4.

Customer: MCC
Designation: 290
Transaction Date: 6/8/2004
Amount: \$100.00
Transaction Type: Check
Check No.: 2000
Create

Sensitive information not shown.

Figure 4. Filled-in Form

Clicking Create will give you the opportunity to review your request as shown in Figure 5.

Microsoft Access

Create the following Sales Receipt in I:\Quickbooks Enterprise Companies\TestDataOnly_SBB.QBW?

Customer: MCC
Designation: 290
Transaction Date: 6/8/2004
Transaction Type: Check
Check Number: 2000
Amount: \$100.00

Yes No

Sensitive information not shown.

Figure 5. User Request Verification

If No is selected, the transaction will be cancelled (Figure 6.). If Yes, the transaction will forward to QuickBooks where it can be found in the Sales Receipts (Figures 7 & 8).

Microsoft Access

Cancelled.

OK

Figure 6. Cancelled Transaction



Figure 7. Transaction Completed.

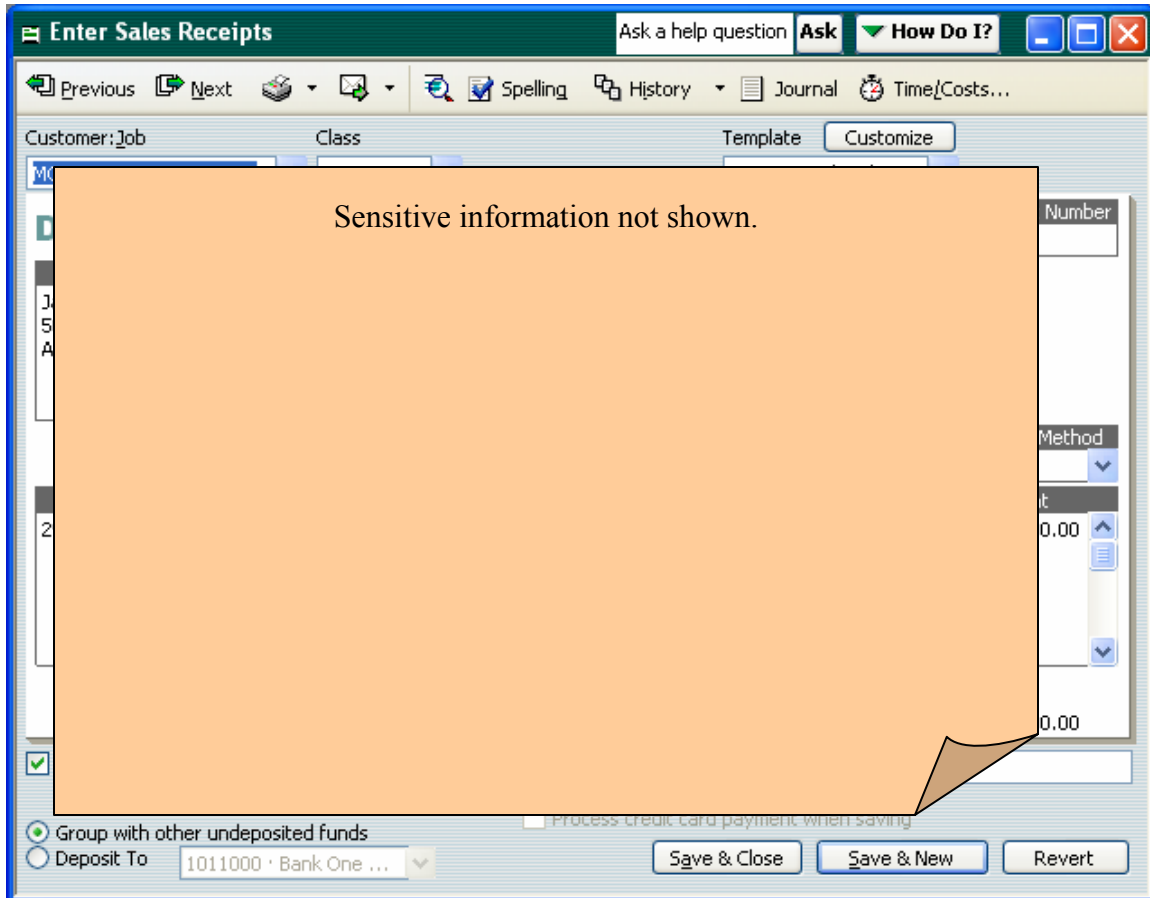


Figure 8. Sales Receipt Created in QuickBooks

The Code

The code behind the form is shown in Appendix A, and includes the button-click subroutine `btnCreateSalesReceipt_Click` and the XML builder function `BuildXML_SalesReceiptAdd`. A high level view of the process of this code follows:

- Verify needed data is provided by the user
- Verify the user wants to create this Sales Receipt
- Build the XML string according to the parameters provided through the form

- Pass the SalesReceiptAdd XML string to QuickBooks

Conclusion

The primary challenge to this study was to create a Sales Receipt Transaction using the QuickBooks SDK, using XML. It is reasonable to conclude that the capability demonstrated in this study provides evidence that automation of Sales Receipt transactions can be achieved with VBA in Microsoft Access, using the QuickBooks SDK.

Appendix A – VBA Code

```
Option Compare Database
Private Sub btnCreateSalesReceipt_Click()

' DESCRIPTION
' Execute button click for adding a Sales Receipt to a QuickBooks file

' HISTORY
' Created by Scott B. Bradley for Cedarstone Partners
' Last Updated June 8, 2004
' Author Contact: s.bradley@computer.org

' OBJECT REFERENCES
' Works with a form containing the following elements:
'   cboCustomer      :   matches Customer:Job in QuickBooks
'   cboDesignation   :   matches full name of Item in QuickBooks
'   txtDate          :   date of transaction in date format
'   cboType           :   payment type: Cash, Check, Money Order, EFT, etc.
'   txtCheckNo       :   Customer's check number
'   txtAmount        :   Dollar amount of transaction as a string

' VBA LIBRARY REFERENCES
'   Microsoft XML, v3.0
'   QBXMLRP2 1.0 Type Library
'   Quickbooks Connector 10.0 Type Library

' REFERENCED FUNCTIONS
'   BuildXML_SalesReceiptAdd by Scott B. Bradley

' REFERENCED SUBROUTINES
'   n/a

' PROCEDURE FLOW
'   1a Check for needed values in the form
'   2a If good to go with needed values, verify with the user
'   3a If user says ok, create the receipt in QuickBooks

' DECLARATIONS
  Dim qbXMLRP As New QBXMLRP2Lib.RequestProcessor2
  Dim strXML As String
  Dim ticket As String
  Dim FS As New FileSystemObject
  Dim strXMLFilePath As String
  Dim stream As TextStream
  Dim strXMLResponse As String

  strQBFile = "I:\Quickbooks Enterprise Companies\TestDataOnly_SBB.QBW"

'
' .....
' .....      btnCreateSalesReceipt_Click      .....
' .....      1a Check for needed values in the form .....
' .....
'
' The needed values are Customer, Designation, Transaction Date, Transaction Type,
Amount
' Check Number is optional and not verified here

Dim strNullCheck As String
strNullCheck = ""

' Customer Check
If IsNull(cboCustomer) Then
  strNullCheck = strNullCheck & "Value required for Customer." & Chr(13) & Chr(13)
End If
```

Creating Sales Receipts in QuickBooks with Access VBA

```

'Designation Check
If IsNull(cboDesignation) Then
    strNullCheck = strNullCheck & "Value required for Designation." & Chr(13) &
Chr(13)
End If

'Transaction Date Check
If IsNull(txtDate) Then
    strNullCheck = strNullCheck & "Value required for Transaction Date." & Chr(13) &
Chr(13)
End If

'Transaction Type Check
If IsNull(cboType) Then
    strNullCheck = strNullCheck & "Value required for Transaction Type." & Chr(13) &
Chr(13)
End If

'Amount Check
If IsNull(txtAmount) Then
    strNullCheck = strNullCheck & "Value required for Amount." & Chr(13) & Chr(13)
End If

If strNullCheck <> "" Then
    MsgBox strNullCheck      'Enlighten the user and
    Exit Sub                'Exit if any needed values are null
End If

'
'
' .....
' .....      btnCreateSalesReceipt_Click .....
' .....      2a If good to go with needed values, verify with the user.....
' .....
' .....

Dim strVerify As String
strVerify = "Create the following Sales Receipt in " & strQBFile & "?" & Chr(13) &
Chr(13) & _
    "Customer: " & cboCustomer & Chr(13) & _
    "Designation: " & cboDesignation & Chr(13) & _
    "Transaction Date: " & txtDate & Chr(13) & _
    "Transaction Type: " & cboType & Chr(13) & _
    "Check Number: " & txtCheckNo & Chr(13) & _
    "Amount: " & Format(txtAmount, "Currency")

Dim intYesNo As Integer
intYesNo = MsgBox(strVerify, vbYesNo)

'
'
' .....
' .....      btnCreateSalesReceipt_Click .....
' .....      3a If user says ok, create the receipt in QuickBooks .....
' .....
' .....

If intYesNo = vbYes Then

    If IsNull(txtCheckNo) Then
        strXML = BuildXML_SalesReceiptAdd(cboCustomer, cboDesignation, txtAmount,
cboType, txtDate)
    Else
        strXML = BuildXML_SalesReceiptAdd(cboCustomer, cboDesignation, txtAmount,
cboType, txtDate, txtCheckNo)
    End If

    qbXMLRP.OpenConnection "", "Sales Receipt Add Request"
    ticket = qbXMLRP.BeginSession(strQBFile, qbFileOpenDoNotCare)

    strXMLResponse = qbXMLRP.ProcessRequest(ticket, strXML)
    qbXMLRP.EndSession ticket
    qbXMLRP.CloseConnection

```

Creating Sales Receipts in QuickBooks with Access VBA

```
MsgBox "Sales Receipt Created"
Exit Sub

Else

    MsgBox "Cancelled."
    Exit Sub          'Exit if user says no

End If

End Sub          'End btnCreateSalesReceipt_Click

Private Function BuildXML_SalesReceiptAdd(strCustomer, strDesignation, strAmount, strType
As String, dtTxnDate As Date, Optional strCheckNo As String) As String

' DESCRIPTION
'   Return qbXML 3.0 string appropriate for passing to QuickBooks for the creation of a
Sales Receipt

' NOTE
'   This function currently has provisions only for specifying full name identifiers, not
list IDs

' HISTORY
'   Created by Scott B. Bradley for Cedarstone Partners
'   Last Updated June 8, 2004
'   Author Contact: s.bradley@computer.org

' FUNCTION PARAMETERS
'   strCustomer      :   Matches Customer:Job in QuickBooks
'   strDesignation   :   Matches full name of Item in QuickBooks
'   strAmount        :   Dollar amount of transaction as a string
'   strType          :   Payment type: Cash, Check, Money Order, EFT, etc.
'   strCheckNo       :   Customer's check number
'   dtTxnDate        :   Date of transaction in date format

' VBA LIBRARY REFERENCES
'   n/a - This function does not pass or execute XML, it only builds the string

' REFERENCED FUNCTIONS
'   n/a

' REFERENCED SUBROUTINES
'   n/a

' PROCEDURE FLOW
'   Build and concat XML components to result in the following string to return
'   (** denotes parameters passed to function):

'   <?xml version="1.0"?><?qbxml version="3.0"?>
'   <QBXML>
'   <QBXMLMsgsRq onError="continueOnError">
'   <SalesReceiptAddRq>
'   <SalesReceiptAdd>
'       <CustomerRef>
'           <FullName>**strCustomer**</FullName>
'       </CustomerRef>
'       <ClassRef>
'           <FullName>**strDesignation**</FullName>
'       </ClassRef>
'       <TemplateRef>
'           <FullName>Custom Cash Sale</FullName>
'       </TemplateRef>
'       <TxnDate>**dtTxnDate**</TxnDate>
'       <CheckNo>**strCheckNo**</CheckNo>
'       <PaymentMethodRef>
'           <FullName>**strType**</FullName>
'       </PaymentMethodRef>
'       <SalesReceiptLineAdd>
'           <ItemRef>
```

Creating Sales Receipts in QuickBooks with Access VBA

```

'           <FullName>**strDesignation**</FullName>
'           </ItemRef>
'           <Amount>**strAmount**</Amount>
'       </SalesReceiptLineAdd>
'   </SalesReceiptAdd>
' </SalesReceiptAddRq>
' </QBXMLMsgsRq>
' </QBXML>

' DECLARATIONS
    Dim xmlCustomerRef, xmlClassRef, xmlTemplateRef, xmlTxnDate, xmlCheckNo,
xmlPaymentMethodRef As String
    Dim xmlSalesReceiptLineAdd, xmlItemRef, xmlAmount As String
'
'CustomerRef Build
xmlCustomerRef = "<CustomerRef>" & _
    "<FullName>" & Replace(strCustomer, "&", "&amp;") & "</FullName>" & _
    "</CustomerRef>"
'
'ClassRef Build
xmlClassRef = "<ClassRef>" & _
    "<FullName>" & Replace(strDesignation, "&", "&amp;") & "</FullName>" & _
    "</ClassRef>"
'
'TemplateRef Build
'The template is hard-coded as below. Your application might require a different template
name
xmlTemplateRef = "<TemplateRef>" & _
    "<FullName>Custom Cash Sale</FullName>" & _
    "</TemplateRef>"
'
'TxnDate Build
xmlTxnDate = "<TxnDate>" & Format(dtTxnDate, "yyyy-mm-dd") & "</TxnDate>"
'
'CheckNumber Build
xmlCheckNo = "<CheckNumber>" & strCheckNo & "</CheckNumber>"
'
'PaymentMethodRef Build
xmlPaymentMethodRef = "<PaymentMethodRef>" & _
    "<FullName>" & strType & "</FullName>" & _
    "</PaymentMethodRef>"
'
'SalesReceiptLineAdd Build
    xmlItemRef = "<ItemRef>" & _
        "<FullName>" & Replace(strDesignation, "&", "&amp;") & "</FullName>" & _
        "</ItemRef>"

    xmlAmount = "<Amount>" & Format(strAmount, "Standard") & "</Amount>"

xmlSalesReceiptLineAdd = "<SalesReceiptLineAdd>" & _
    xmlItemRef & xmlAmount & _
    "</SalesReceiptLineAdd>"
'
'Put it all together
BuildXML_SalesReceiptAdd = "<?xml version=""1.0""?><?qbxml version=""3.0""?>" & _
    "<QBXML>" & _
    "<QBXMLMsgsRq onError=""continueOnError"">" & _
    "<SalesReceiptAddRq>" & _
    "<SalesReceiptAdd>" & _
        xmlCustomerRef & xmlClassRef & xmlTemplateRef & xmlTxnDate
& _
        xmlCheckNo & xmlPaymentMethodRef & xmlSalesReceiptLineAdd
& _
    "</SalesReceiptAdd>" & _

```

Creating Sales Receipts in QuickBooks with Access VBA

```
        "</SalesReceiptAddRq>" & _  
        "</QBXMLMsgsRq>" & _  
        "</QBXML>"  
End Function      'End BuildXML_SalesReceiptAdd
```