Session: 45TC Spring COMMON (404384)

iSeries Access ActiveX Development

IBM @server iSeries

Troy C. Bleeker
bleek@us.ibm.com

Agenda

IBM @server iSeries

- Goals
- Interfaces
  - ADO/OLE DB, iSeries Access ActiveX Objects, iSeries Access ActiveX Controls
- Getting Started
  - Installation
  - Documentation
  - Samples
  - Project References and Components
- Demonstration and Code Walk-thru
  - iSeries Access Toolkit Visual Basic Wizards
  - iSeries Access ActiveX Objects

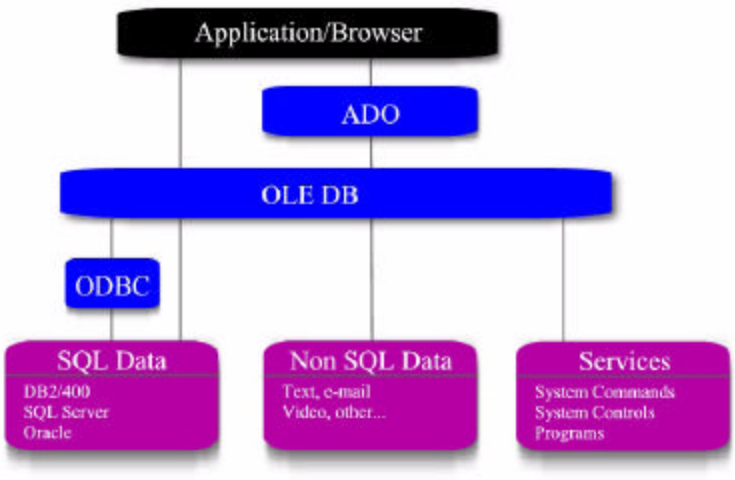
Goals

iSeries Access ActiveX Programming Goals

- Make client/server application development easy
- Work with popular Windows 9x/Me/NT/2000 products
  - Visual Basic, PowerBuilder, Delphi, ...
  - Microsoft Office, Lotus Notes, ...
- Simple, consistent interfaces
  - Microsoft ActiveX objects
- Enhanced functionality each release
- Performance >= what you are used to

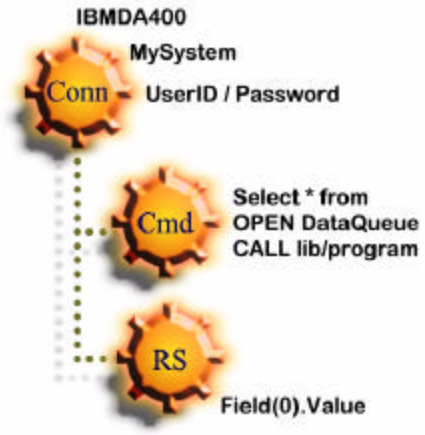
ADO/OLE DB Interfaces

Overview



ActiveX Data Object (ADO) Model

IBM server iSeries



iSeries Access OLE DB Provider (IBMDA400)

IBM server iSeries

- Functionality
  - Connections
  - Tables record level access
  - SQL statements
  - SQL stored procedures
  - Data queues
  - CL commands
  - Program call
  - Error information

IBM @server. For the next generation of e-business.

OLE DB Provider for ODBC Drivers (MSDASQL)

IBM server iSeries

- Use with iSeries Access ODBC driver
  - Must define an ODBC data source
- Functionality
  - SQL statements
  - SQL stored procedures

iSeries Access Toolkit Visual Basic Wizards

IBM server iSeries

- Jump start your ADO/OLE DB development
- Code generating wizards (available for Visual Basic, not .Net)
- Visual Basic Wizards Help

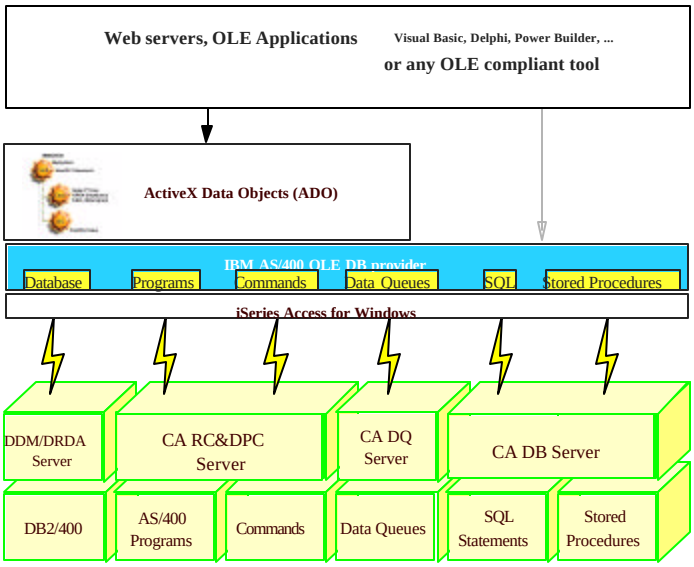
"iSeries ADO Toolkit" VB menu Add-In

- Link Tables...
- Link Stored Procedures...
- Link Data Queues...
- Link Commands...
- Link Programs...
- Create Form from Links...
- Work with Stored Procedures...
- Work with Data Queues...
- Options...
- Help...

IBM @server. For the next generation of e-business.

iSeries Host Servers

IBM server iSeries



IBM server iSeries

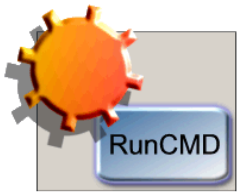
ActiveX Object Interfaces

ActiveX Object Model (two examples of many)

IBM server iSeries

Controls

cwbSystemListBox

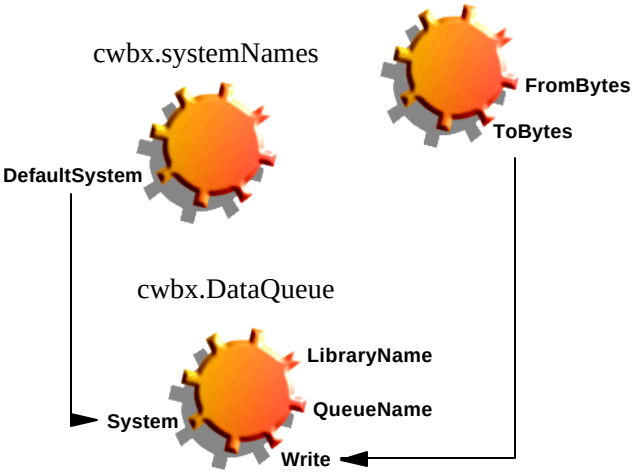


Objects

cwbx.StringConverter

cwbx.systemNames

cwbx.DataQueue



iSeries Access ActiveX Objects

IBM server iSeries

- Client information
- AS/400 system list
- Connections
- Data queues
- CL commands
- Program call
- Error Information
- Numeric conversions
- Structure conversions
- Code page conversions

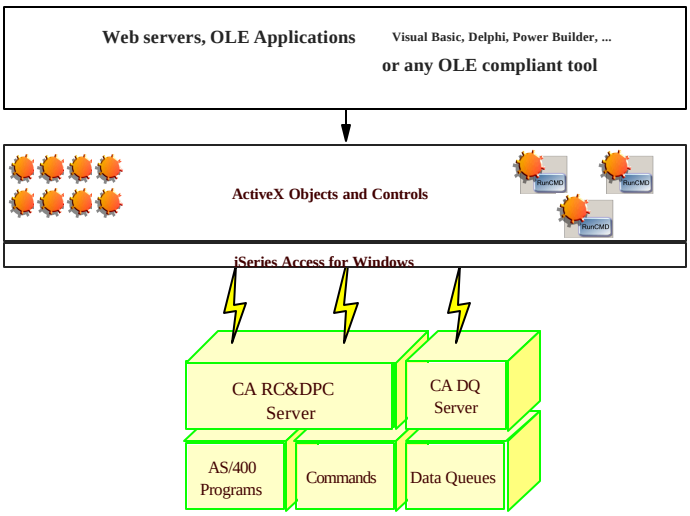
iSeries ActiveX Controls

IBM server iSeries

- AS/400 system list box
- Data queue text box
- Remote command button

iSeries Host Servers

IBM server iSeries



Getting Started

Installation

- iSeries Access Required Programs

  (ActiveX objects ship in here. They don't need to be selected)
- Data Access
  - OLE DB Provider
  - ODBC (for use with MSDASQL only)
- Programmer's Toolkit
  - Documentation
  - Visual Basic Wizards

ADO and OLE DB Documentation

IBM server iSeries

- A Fast Path to AS/400 Client/Server Using AS/400 OLE DB Support (SG24-5183)
  - http://www.redbooks.ibm.com
- iSeries Access OLE DB Technical Reference
  - From the iSeries Access Start Bar menu item
    - Programmer's Toolkit->Common Interfaces->ADO/OLE DB
- Microsoft ADO Help
  - http://www.microsoft.com/data/doc.htm
- iSeries Access ADO/OLE DB Web Page
  - http://www.as400.ibm.com/clientaccess/oledb

iSeries Access ActiveX Object Documentation

IBM server iSeries

- AS/400 Client Access Express for Windows: Implementing V4R4M0 (SG24-5191)
  - http://www.redbooks.ibm.com
- iSeries Access ActiveX Automation Objects
  - From the iSeries Access Start Bar menu item
    - Programmer's Toolkit->Programming Technologies->ActiveX
- AS/400 System API Reference (SC41-5801)
- OS/400 CL Reference (SC41-5722)

Program and Code Samples

IBM server iSeries

- ADO/OLE DB
  - Download from
    www.ibm.com/servers/eserver/iseries/clientaccess/oledb/samples.htm
  - A Fast Path to AS/400 Client/Server Using AS/400 OLE DB Support (SG24-5183)
    - http://www.redbooks.ibm.com
    - Download from "Additional materials"
- iSeries Access ActiveX Objects
  - Download from
    www.ibm.com/servers/eserver/iseries/clientaccess/toolkit/activex.htm

Project References and Components

IBM server iSeries

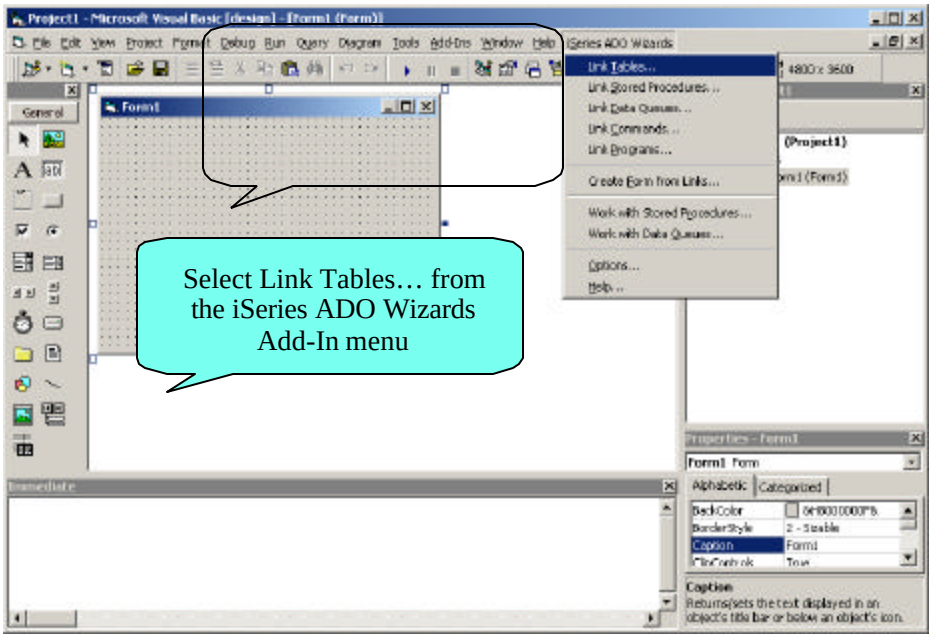
- Project references
  - ADO/OLE DB
    - Microsoft ActiveX Data Objects 1.5 Library (msado15.dll)
    - *AS/400 Express Toolkit Table Index Type Library 1.2 (cwbzzidx.dll)*
  - iSeries Access ActiveX Objects
    - IBM AS/400 Client Access Express ActiveX Object Library (cwbx.dll)
- Project components
  - iSeries Access ActiveX Controls
    - Client Access Control Library (cwbctrl.ocx)

Available, but deprecated after V4R5M0

Demo and Code

iSeries Access Toolkit Visual Basic Wizards

IBM server iSeries



IBM @server. For the next generation of e-business.

Visual Basic Add-Ins (code)

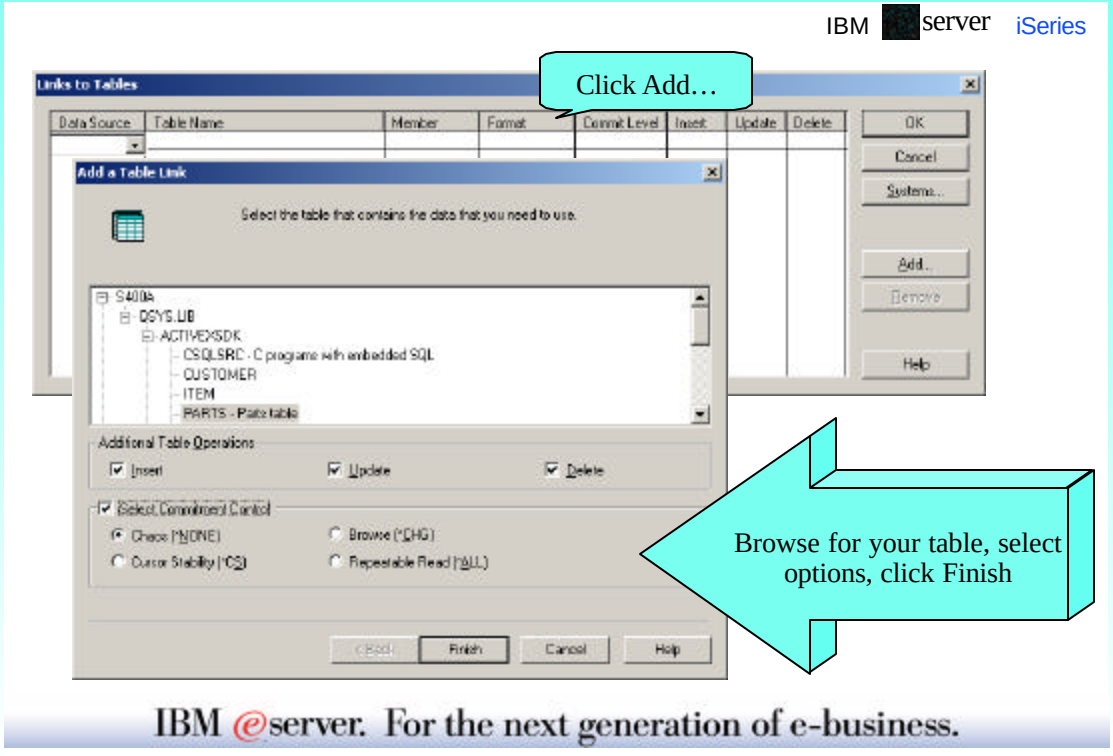
Form1.frm

```
'{{DA400_LINKS_BEGIN}}
Public Links As DA400Links
'{{DA400_LINKS_END}}

Dim EndOfDeclares As String 'keeps comments from floating
Private Sub Form_Load()
'{{DA400_LINK_BEGIN}}
Set Links = New DA400Links
'{{DA400_LINK_END}}
End Sub
```

DA400Links.cls

```
'NOTE - These calls initialize connections and prepared commands
'{{DA400_INIT_BEGIN}}
Call Connect
Call Prepare
Call OpenLinks
'{{DA400_INIT_END}}
End Sub
```



Add a Table Link (code)

DA400Links.cls

```
'NOTE - The Toolkit will add and remove code between markers.  Do not comment or delete the
      lines between the markers.
Option Explicit
'{{DA400_CONNECTIONS_BEGIN}}
Public cnS400A As New ADODB.Connection
'{{DA400_CONNECTIONS_END}}

'{{DA400_TABLES_BEGIN}}
Public rs_ACTIVEXSDK_PARTS As New ADODB.Recordset
'{{DA400_TABLES_END}}

Public Sub Connect()
'NOTE - The Toolkit will add and remove code between markers.  Do not comment or delete the
      lines between the markers.
'{{DA400_CONNECTS_BEGIN}}
cnS400A.Open "Provider=IBMDA400;Data Source=S400A;", "", ""s
'{{DA400_CONNECTS_END}}
End Sub
```

Previous to V5R1M0 an index object was used. It is no longer used.

Add a Table Link (code)

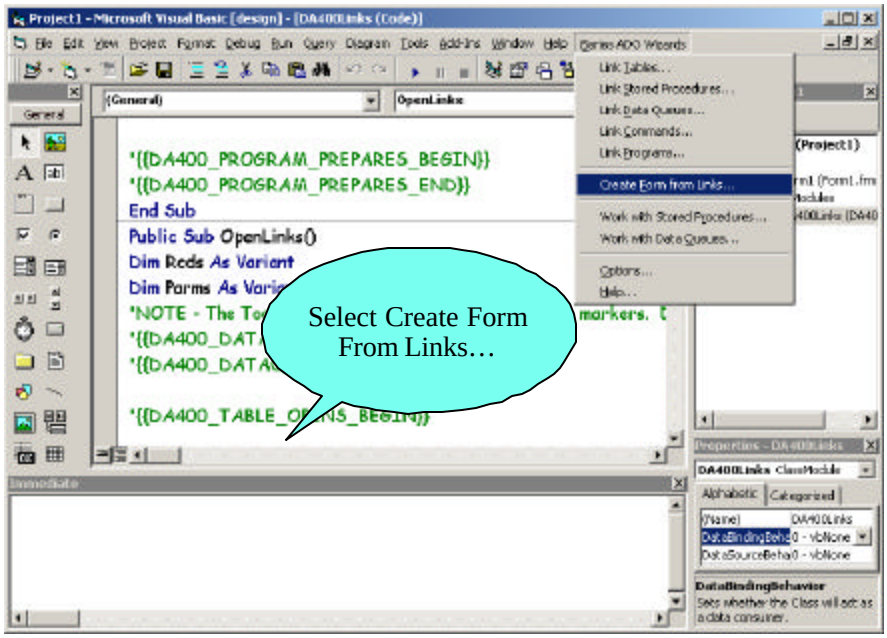
DA400Links.cls

```
Public Sub OpenLinks()
Dim Rcds As Variant
Dim Parms As Variant
'NOTE - The Toolkit will add and remove code between markers.  Do not comment or delete the
      lines between the markers.
'{{DA400_DATAQUEUE_OPENS_BEGIN}}
'{{DA400_DATAQUEUE_OPENS_END}}

'{{DA400_TABLE_OPENS_BEGIN}}
rs_ACTIVEXSDK_PARTS.Index = "/QSYS.LIB/ACTIVEXSDK.LIB/PARTS.FILE(*FIRST, *NONE)"
rs_ACTIVEXSDK_PARTS.CursorLocation = adUseServer
rs_ACTIVEXSDK_PARTS.Open "/QSYS.LIB/ACTIVEXSDK.LIB/PARTS.FILE(*FIRST, *NONE)",
      cnS400A, adOpenDynamic, adLockOptimistic, adCmdTableDirect
'{{DA400_TABLE_OPENS_END}}
End Sub
```

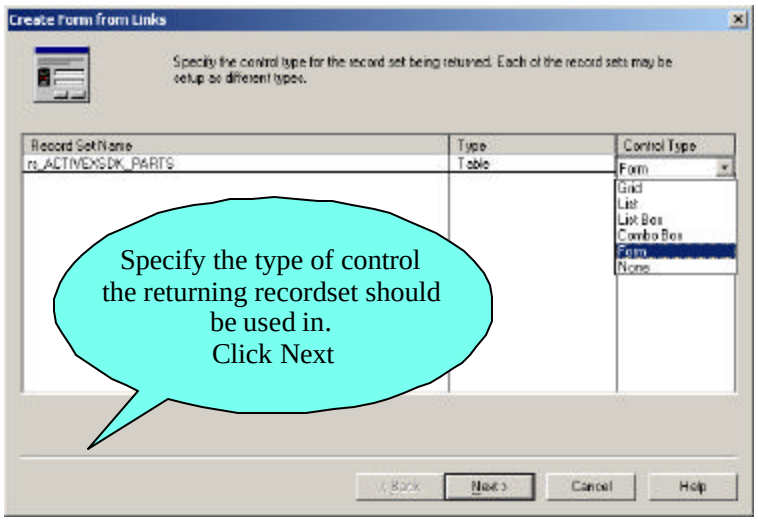
Create a form from the link

IBM server iSeries



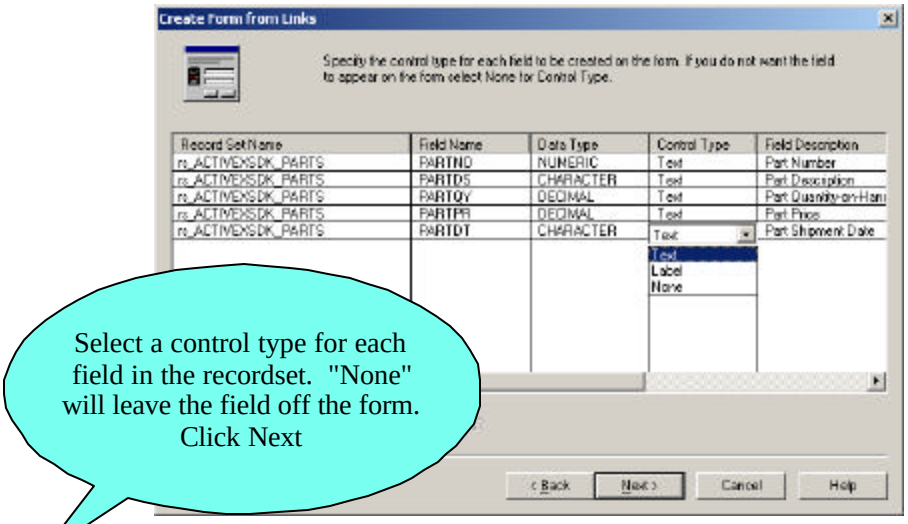
Recordset as a form

IBM server iSeries



Control types for each field

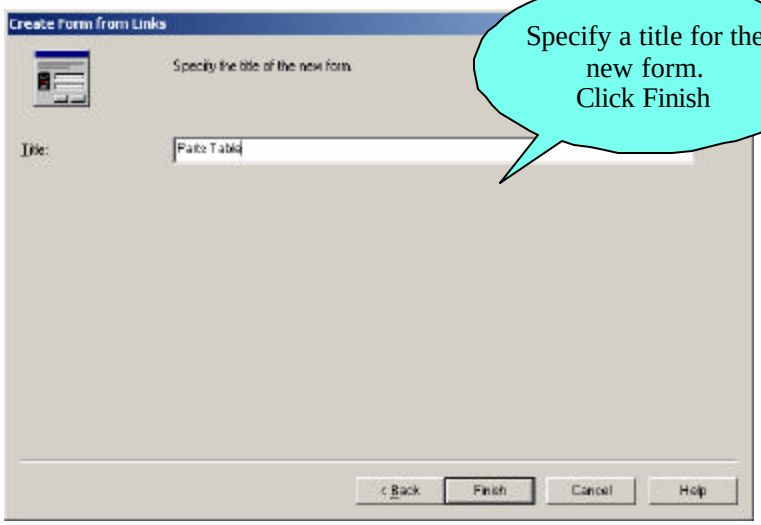
IBM server iSeries



Select a control type for each field in the recordset. "None" will leave the field off the form. Click Next

New form title

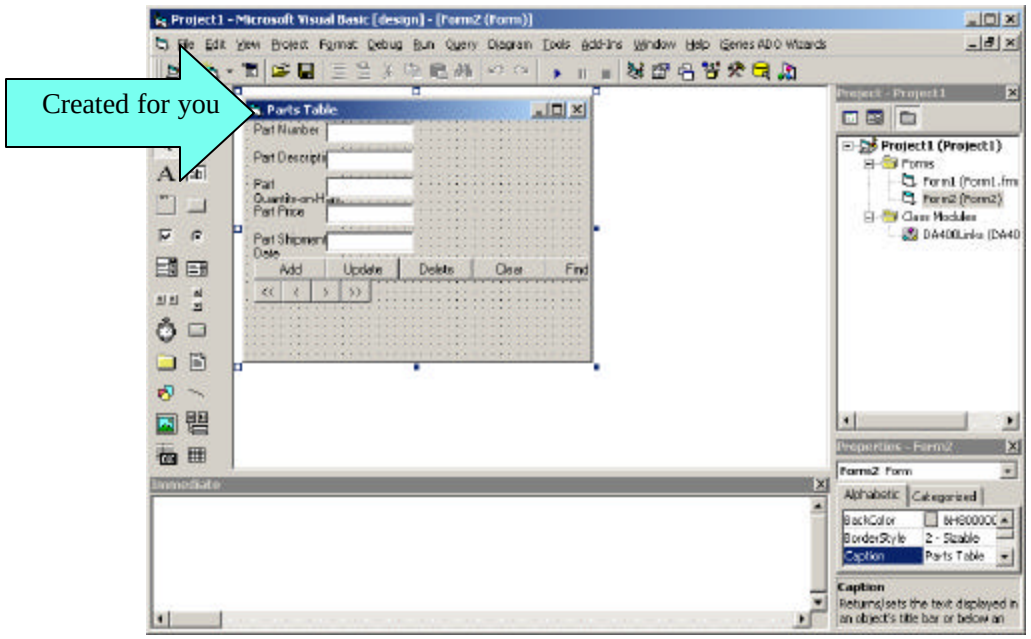
IBM server iSeries



Specify a title for the new form. Click Finish

The new form, Form2

IBM server iSeries



IBM @server. For the next generation of e-business.

The new form, Form2 (Load and Fill_Form code)

Form2.frm

```
Dim Links As DA400Links

Private Sub Form_Load()
Dim Rcds As Variant
Dim Parms As Variant
Dim fso
Dim blobData() As Byte

Set Links = Form1.Links
  Fill_Form
End Sub

Private Sub Fill_Form()
Dim val As Variant
  If Not Links.rs_ACTIVEXSDK_PARTS.BOF And Not Links.rs_ACTIVEXSDK_PARTS.EOF Then
    val = Links.rs_ACTIVEXSDK_PARTS.Fields(0).Value
    If VarType(val) = vbNull Then
      Text1.Text = "<NULL>"
    Else
      Text1.Text = val
    End If
```

The new form, Form2 (Fill_Form code *continued*)

Form2.frm

```
    val = Links.rs_ACTIVEXSDK_PARTS.Fields(1).Value
    If VarType(val) = vbNull Then
      Text2.Text = "<NULL>"
    Else
      Text2.Text = val
    End If
    val = Links.rs_ACTIVEXSDK_PARTS.Fields(2).Value
    If VarType(val) = vbNull Then
      Text3.Text = "<NULL>"
    Else
      Text3.Text = val
    End If
    val = Links.rs_ACTIVEXSDK_PARTS.Fields(3).Value
    If VarType(val) = vbNull Then
      Text4.Text = "<NULL>"
    Else
      Text4.Text = val
    End If
    val = Links.rs_ACTIVEXSDK_PARTS.Fields(4).Value
    If VarType(val) = vbNull Then
      Text5.Text = "<NULL>"
    Else
      Text5.Text = val
    End If
  Else
      MsgBox "You have reached the beginning or end of the file.", vbInformation
      ClearBtn_Click
  End If
End Sub
```

The new form, Form2 (Add button code)

Form2.frm

```
Private Sub AddBtn_Click()
Dim Flds As Variant
Dim Vals As Variant

  Flds = Array("PARTNO", "PARTDS", "PARTQY", "PARTPR", "PARTDT")
  Vals = Array(Text1.Text, Text2.Text, Text3.Text, Text4.Text, Text5.Text)
  For I = LBound(Vals) To UBound(Vals)
    If Vals(I) = "<NULL>" Then Vals(I) = Null
  Next I
  Links.rs_ACTIVEXSDK_PARTS.AddNew Flds, Vals
  MsgBox "The current record was added.", vbInformation
End Sub
```

The new form, Form2 (Update button code)

Form2.frm

```
Private Sub UpdateBtn_Click()
Dim Flds As Variant
Dim Vals As Variant

  If Not Links.rs_ACTIVEXSDK_PARTS.BOF And Not Links.rs_ACTIVEXSDK_PARTS.EOF Then
      Flds = Array("PARTNO", "PARTDS", "PARTQY", "PARTPR", "PARTDT")
      Vals = Array(Text1.Text, Text2.Text, Text3.Text, Text4.Text, Text5.Text)
      For I = LBound(Vals) To UBound(Vals)
        If Vals(I) = "<NULL>" Then Vals(I) = Null
      Next I
      Links.rs_ACTIVEXSDK_PARTS.Update Flds, Vals
  Else
     MsgBox "You are currently not positioned on a record.  Press next or previous button to be
      positioned at a record.", vbInformation
  End If
End Sub
```

The new form, Form2 (Delete and Clear button code)

Form2.frm

```
Private Sub DeleteBtn_Click()
  If Not Links.rs_ACTIVEXSDK_PARTS.BOF And Not Links.rs_ACTIVEXSDK_PARTS.EOF Then
      Links.rs_ACTIVEXSDK_PARTS.Delete adAffectCurrent
      MsgBox "The current record was deleted.", vbInformation
      ClearBtn_Click
  Else
      MsgBox "You are currently not positioned on a record.  Press next or previous button to be
      positioned at a record.", vbInformation
  End If
End Sub

Private Sub ClearBtn_Click()
  Text1.Text = ""
  Text2.Text = ""
  Text3.Text = ""
  Text4.Text = ""
  Text5.Text = ""
End Sub
```

The new form, Form2 (Find button code)

Form2.frm

```
Private Sub FindBtn_Click()
Dim Keys As Variant
Dim Key1 As Variant

  Key1 = InputBox("Enter value for key PARTNO.")
  Keys = Array(Key1)
  On Error GoTo SeekFailed
  Links.rs_ACTIVEXSDK_PARTS.Seek Keys, adSeekFirstEQ
  Fill_Form
  Exit Sub
SeekFailed:
  MsgBox "Record not found for the key value that you specified.", vbInformation
End Sub
```

The new form, Form2 (First and Previous button code)

Form2.frm

```
Private Sub FirstBtn_Click()
  If (Links.rs_ACTIVEXSDK_PARTS.EOF And Links.rs_ACTIVEXSDK_PARTS.BOF) Then
      MsgBox "The file is empty.", vbInformation
  Else
      Links.rs_ACTIVEXSDK_PARTS.MoveFirst
      Fill_Form
  End If
End Sub

Private Sub PreviousBtn_Click()
  If Links.rs_ACTIVEXSDK_PARTS.BOF Then
       MsgBox "You have reached the beginning of the file.", vbInformation
  Else
       Links.rs_ACTIVEXSDK_PARTS.MovePrevious
       If Links.rs_ACTIVEXSDK_PARTS.BOF Then
         MsgBox "You have reached the beginning of the file.", vbInformation
         Links.rs_ACTIVEXSDK_PARTS.MoveNext
       Else
         Fill_Form
       End If
  End If
End Sub
```

The new form, Form2 (Next and Last button code)

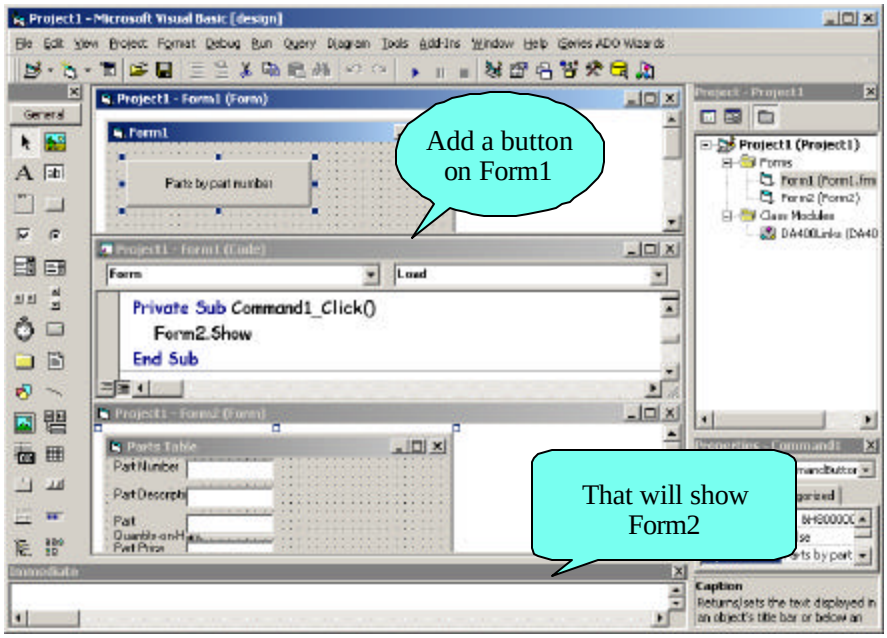
Form2.frm

```
Private Sub NextBtn_Click()
  If Links.rs_ACTIVEXSDK_PARTS.EOF Then
     MsgBox "You have reached the end of the file.", vbInformation
  Else
     Links.rs_ACTIVEXSDK_PARTS.MoveNext
     If Links.rs_ACTIVEXSDK_PARTS.EOF Then
        MsgBox "You have reached the end of the file.", vbInformation
        Links.rs_ACTIVEXSDK_PARTS.MovePrevious
     Else
        Fill_Form
     End If
  End If
End Sub

Private Sub LastBtn_Click()
  If (Links.rs_ACTIVEXSDK_PARTS.EOF And Links.rs_ACTIVEXSDK_PARTS.BOF) Then
       MsgBox "The file is empty.", vbInformation
  Else
       Links.rs_ACTIVEXSDK_PARTS.MoveLast
       Fill_Form
  End If
End Sub
```

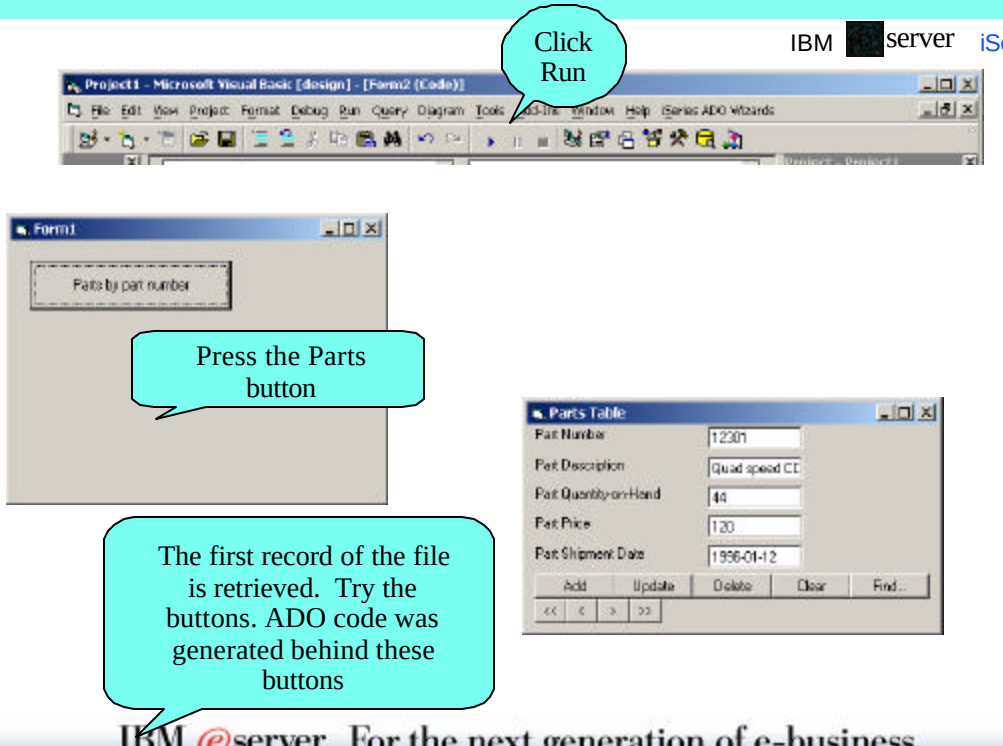
Putting it all together

IBM server iSeries



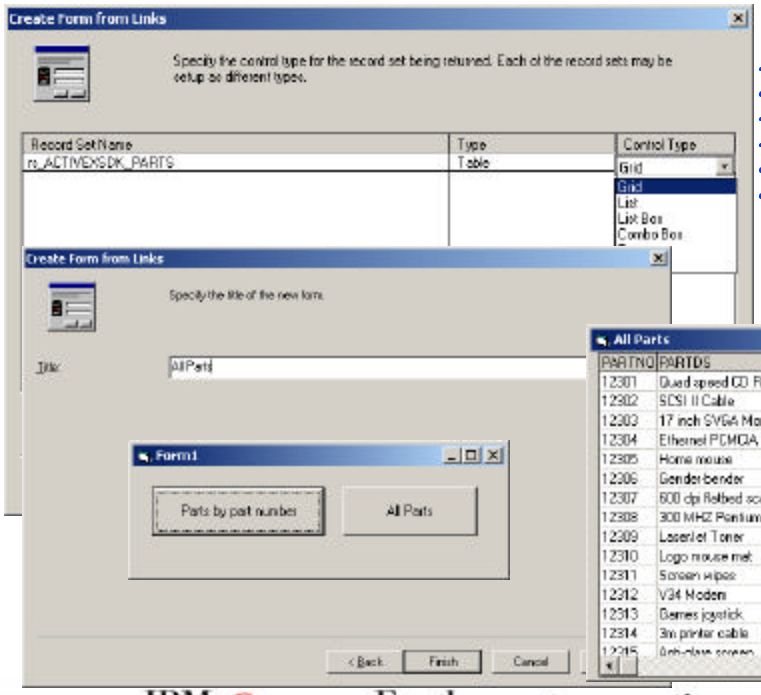
IBM @server. For the next generation of e-business.

Run the new application



IBM @server. For the next generation of e-business.

To get all the data in a grid



- Select Create Form from Links
- Specify a Grid control type
- Specify new form title
- Size the grid to fit the form
- Add a new button to show Form3
- Run the application

IBM @server. For the next generation of e-business.

IBM server iSeries

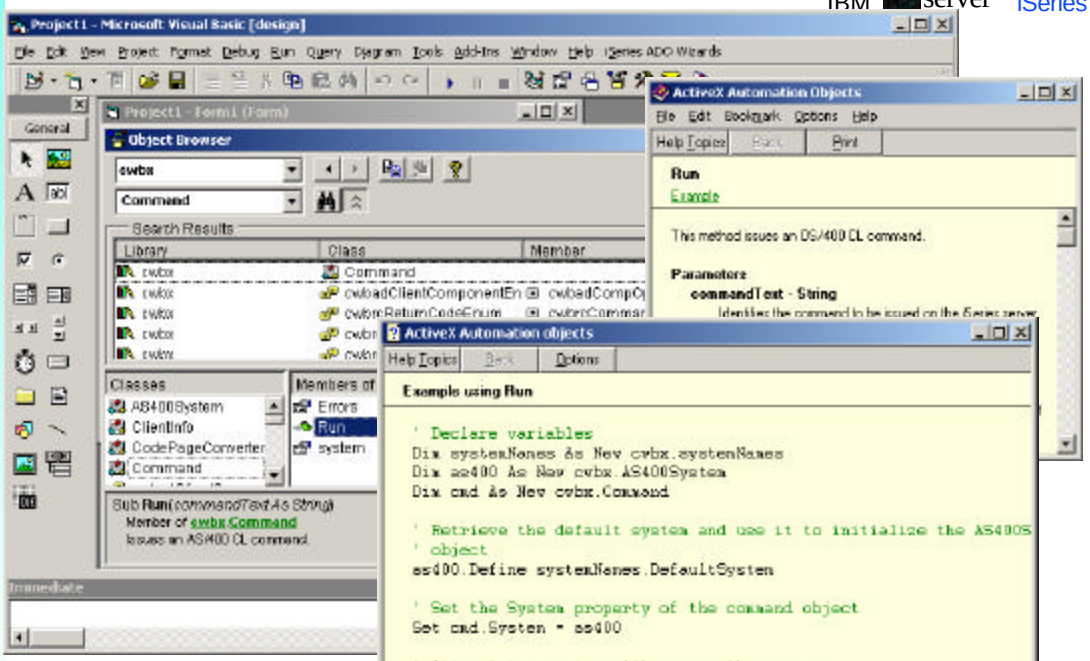
Demo and Code

iSeries Access ActiveX Objects

IBM @server. For the next generation of e-business.

View the Object Browser

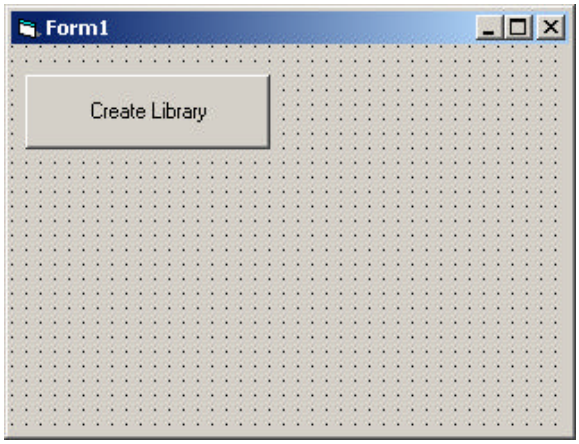
IBM server iSeries



IBM @server.

Create Library

IBM server iSeries



Create Library (code)

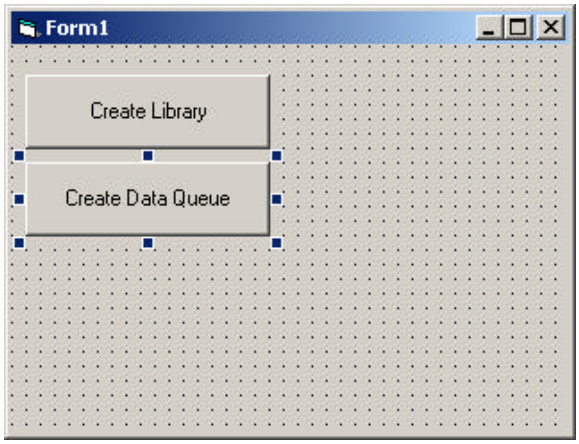
Form1.frm

Set the iSeries to use as the default system.

```
Private Sub Command1_Click()
    ' Declare variables
    Dim systemNames As New cwbx.systemNames
    Dim as400 As New cwbx.AS400System
    Dim cmd As New cwbx.Command
    ' Retrieve the default system and use it to initialize the AS400System
    ' object
    as400.Define systemNames.DefaultSystem
    ' Set the System property of the command object
    Set cmd.System = as400
    ' Attempt to create a library on the server
        cmd.Run "crtlib myLib text('My New Library')"
        MsgBox "Library MYLIB created."
End Sub
```

Create Data Queue

IBM server iSeries



Create Data Queue (code)

Form1.frm

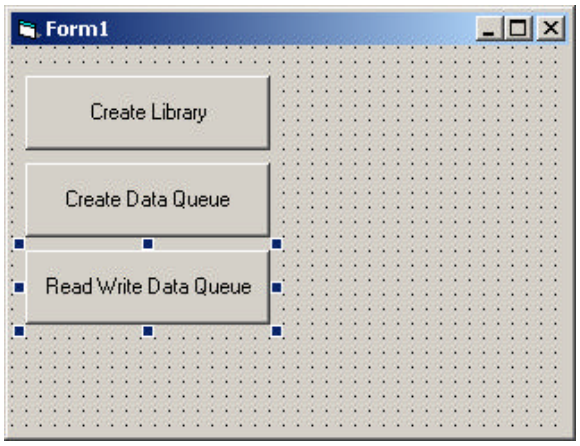
```
Private Sub Command2_Click()
    ' Declare variables
    Dim systemNames As New cwbx.systemNames
    Dim as400 As New cwbx.AS400System
    Dim dq As New cwbx.DataQueue
    Dim dqa As New cwbx.DataQueueAttributes
    ' Retrieve the default system and use it to initialize the AS400System
    ' object
    as400.Define systemNames.DefaultSystem
    ' Set the System property of the DataQueue object
    Set dq.System = as400
    ' Set the LibraryName property.  Note: This library needs to exist
    ' before attempting to create the data queue on the server. If the
    ' library does not exist, the query of the Exists property will fail
    ' with a cwbdqLibraryNotFound error (The Command object can be used
    ' to create the library)
    dq.LibraryName = "myLib"
    ' Set the QueueName property
    dq.QueueName = "myQ"
```

Create Data Queue (code *continued*)

Form1.frm

```
    ' Check to see if the data queue already exists on the server
    If (dq.Exists = False) Then
      ' Set properties in the DataQueueAttributes object to override
      ' some of the default values
      dqa.Description = "My new server data queue"
      dqa.MaxRecordLength = 100
      dqa.RetrievalOrder = cwbdqSeqLifo
       dqa.SenderInfoSaved = True
      ' Create the data queue on the server, using the attributes set
      ' in the DataQueueAttributes object. If a DataQueueAttributes
      ' was not passed, the default attribute values would have been
      ' used
      dq.Create dqa
      MsgBox "Data queue MYQ created."
    End If
End Sub
```

Read/Write Data Queue



Read/Write Data Queue (code)

Form1.frm

```
Private Sub Command3_Click()
    ' Declare variables
    Dim systemNames As New cwbx.systemNames
    Dim as400 As New cwbx.AS400System
    Dim dq As New cwbx.DataQueue
    Dim stringCvtr As New cwbx.StringConverter
    ' Retrieve the default system and use it to initialize the AS400System
    ' object
    as400.Define systemNames.DefaultSystem
    ' Set the System property of the DataQueue object
    Set dq.System = as400
    ' Set the LibraryName property.
    dq.LibraryName = "myLib"
    ' Set the QueueName property (Assume this queue exists on the server)
    dq.QueueName = "myQ"
    ' Write some strings to the server data queue. The StringConverter
    ' object is used to convert the string to a byte array
    dq.Write stringCvtr.ToBytes("String 1")
    dq.Write stringCvtr.ToBytes("String 2")
    dq.Write stringCvtr.ToBytes("String 3")
    dq.Write stringCvtr.ToBytes("String 4")
```

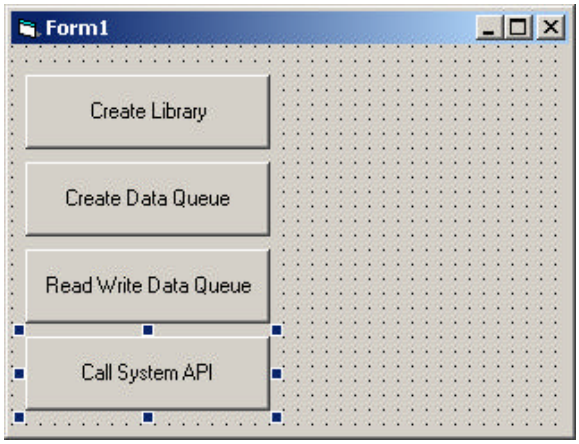
Read/Write Data Queue (code *continued*)

Form1.frm

```
    ' Set the MaximumRetrievalLength property, since we know none of
    ' the records on the queue are very long. This saves memory
    dq.MaximumRetrievalLength = 25
    ' Peek the first string. This does not remove the record from the
    ' queue. The StringConverter object is used to convert the string
    ' from a byte array to a string
    MsgBox "Record peeked = " & stringCvtr.FromBytes(dq.Peek)
    ' Read the first string. This removes the record from the queue.
    MsgBox "Record read = " & stringCvtr.FromBytes(dq.Read)
    ' Read the second string.
    MsgBox "Record read = " & stringCvtr.FromBytes(dq.Read)
    ' Clear the rest of the records from the queue
    dq.Clear
End Sub
```

Call System API

IBM server iSeries



Call System API (code)

Form1.frm

```
Private Sub Command4_Click()
    ' Declare variables
    Dim systemNames As New cwbx.systemNames
    Dim as400 As New cwbx.AS400System
    Dim GetLibraryDesc As New cwbx.Program
    Dim parms As New cwbx.ProgramParameters
    Dim strCvtr As New cwbx.StringConverter
    Dim infoRequested As New cwbx.Structure
    Dim longCvtr As New cwbx.LongConverter
    Dim libraryInfo As New cwbx.Structure
    ' Retrieve the default system and use it to initialize the AS400System
    ' object
    as400.Define systemNames.DefaultSystem
    ' Set the System property of the Program object
    Set GetLibraryDesc.System = as400
    ' Set the LibraryName property of the Program object
    GetLibraryDesc.LibraryName = "QSYS"
    ' Set the ProgramName property of the Program object
    GetLibraryDesc.ProgramName = "QLIRLIBD"
```

Call System API (code *continued*)

Form1.frm

```
'Define parameters and set all input parameter values
parms.Append "library info", cwbrcOutput, 78
parms.Append "library info size", cwbrcInput
parms("library info size") = longCvtr.ToBytes(78)
' Pad the library name with blanks out to 10 character
strCvtr.Length = 10
parms.Append "library name", cwbrcInput
parms("library name") = strCvtr.ToBytes("MYLIB")
' Request just the library description
parms.Append "info requested", cwbrcInput
infoRequested.Fields.Append "number of keys", 4
infoRequested.Fields("number of keys") = longCvtr.ToBytes(1)
infoRequested.Fields.Append ("request description"), 4
infoRequested.Fields("request description") = longCvtr.ToBytes(5)
parms("info requested") = infoRequested.Bytes
parms.Append "error code", cwbrcInput
parms("error code") = longCvtr.ToBytes(0)
```

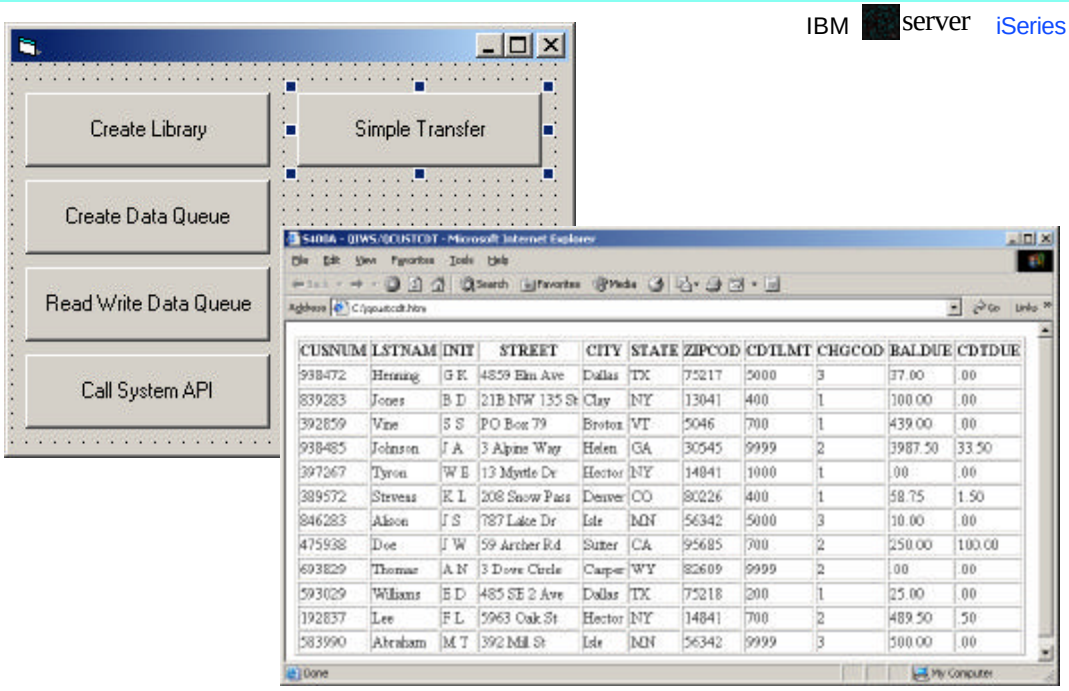
Call System API (code *continued*)

Form1.frm

```
' Call the system API to retrieve the library description
GetLibraryDesc.Call parms
' Store the library information in a Structure object so
' we can retrieve individual field values
libraryInfo.Bytes = parms("library info").Value
' Define the layout of the returned message information
' There is a nested structure within this structure, but we will
' just define it as one flat structure
libraryInfo.Fields.Append "bytesReturned", 4
libraryInfo.Fields.Append "bytesAvailable", 4
libraryInfo.Fields.Append "variableLengthReturned", 4
libraryInfo.Fields.Append "variableLengthAvailable", 4
libraryInfo.Fields.Append "lengthReturned", 4
libraryInfo.Fields.Append "requestKey", 4
libraryInfo.Fields.Append "fieldSize", 4
libraryInfo.Fields.Append "fieldValue", _
  longCvtr.FromBytes(libraryInfo("fieldSize").Value)
' Display the retrieved library description
strCvtr.Length = 0
MsgBox "Description of MYLIB:" & vbCr & _
  strCvtr.FromBytes(libraryInfo("fieldValue").Value)
End Sub
```

Data Transfer Simple

IBM server iSeries



CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37.00	.00
839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100.00	.00
392859	Vine	S S	PO Box 79	Broton	VT	5046	700	1	439.00	.00
938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.50	33.50
397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	.00	.00
389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	1.50
846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10.00	.00
475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250.00	100.00
693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	.00	.00
593029	Wilkins	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25.00	.00
192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.50	.50
583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500.00	.00

IBM @server. For the next generation of e-business.

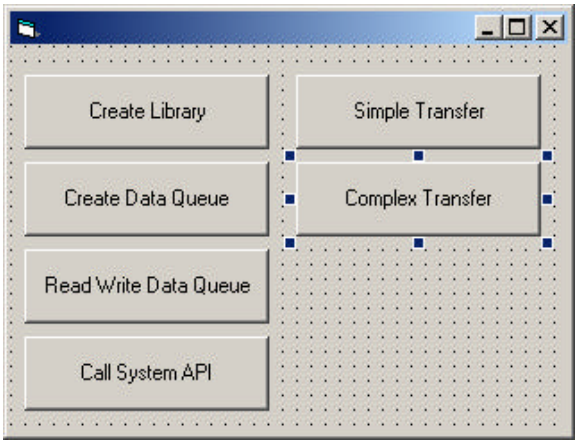
Data Transfer Simple (code)

Form1.frm

```
Private Sub Command5_Click()
    ' Declare variables
    Dim dt As New cwbx.DatabaseTransfer
    Dim systemNames As New cwbx.systemNames
    ' Download QIWS/QCUSTCDT from the default server, to the C: drive
    ' and store it in HTML
    dt.Download systemNames.DefaultSystem, "qiws/qcustcdt", "c:\qcustcdt.htm", cwbdtHTML
End Sub
```

Data Transfer Complex

IBM server iSeries



IBM @server. For the next generation of e-business.

Data Transfer Complex (code)

Form1.frm

```
Private Sub Command6_Click()
    ' Declare variables
    Dim systemNames As New cwbx.systemNames
    Dim dt As New cwbx.DatabaseTransfer
    ' The following variables are only needed for setup, not for the transfer itself
    Dim as400 As New cwbx.AS400System
    Dim dlr As New cwbx.DatabaseDownloadRequest

    ' Retrieve the default system and use it to initialize the AS400System
    ' object
    as400.Define systemNames.DefaultSystem
    ' Set the System property of the download request object
    Set dlr.System = as400
    ' Set properties to download only the LSTNAM column of QIWS/QCUSTCDT
    dlr.AS400File = "qiws/qcustcdt"
    dlr.pcFile = "c:\qcustlst.txt"
    dlr.Query.Select = "LSTNAM"
    ' Save the transfer request
    dlr.SaveRequest "c:\qcustlst.dtf"
```

Data Transfer Complex (code *continued*)

Form1.frm

```
    ' Set specific security information to be used for the transfer
    dt.UserID = "me"
    dt.Password = "secret"
    On Error Resume Next
    ' Attempt to run the transfer request with the bad security information.
    ' Since both the user ID and password are set, we will not be prompted
    ' for security information
    dt.Transfer "c:\qcustlst.dtf"
    ' Display the error information resulting from the invalid security values passed
    MsgBox "Error Number = " & Err.Number & vbCr & _
    "Error Description = " & vbCr & Err.Description & vbCr & _
    "Error Source = " & Err.Source & vbCr & _
    "iSeries Access for Windows Return Code = " & dt.Errors.ReturnCode
    ' Display any iSeries Access for Windows messages returned
    For Each errMsg In dt.Errors
    MsgBox "iSeries Access for Windows Message Text: " & vbCr & errMsg.Text
    Next
    On Error GoTo 0
    ' Reset the user ID and password so we will be prompted (or cached
    ' information will be used)
    dt.UserID = ""
    dt.Password = ""
    ' Try the transfer request again
    dt.Transfer "c:\qcustlst.dtf"
End Sub
```

Trademarks and Disclaimers

IBM server iSeries

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM
AS/400e	IBM (logo)
eServer	iSeries
	OS/400

SET

Lotus and SmartSuite are trademarks of Lotus Development Corporation and/or IBM Corporation in the United States, other countries, or both.
MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.
Microsoft and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

IBM @server. For the next generation of e-business.

License and Disclaimer

IBM server iSeries