
Internet ActiveX Control

Internet ActiveX Overview

Like the WebClient, the Internet ActiveX, or “iX” as it is known, prints across the Internet to locally selected printers that have been configured as CLIENT DEFINED on the server. This Chapter contains references to applications that are also documented in previous chapters. It is recommended that you read Chapter 3, the Internet Printing chapter, before continuing.

This chapter includes information on the use of Web Servers, JSPs, servlets, and servlet engines, all of which are components of the Internet ActiveX.

Programmers developing in 32-bit languages supporting ActiveX Controls can easily interface the Software Print Server functions with their own applications. In the previous ActiveX chapters, Software discussed the Client Control. These controls may be employed with ease by a user to connect and print to stand-alone and networked bar code label printers. The Internet ActiveX Control (“iX”) can utilize this innovative technology as well.

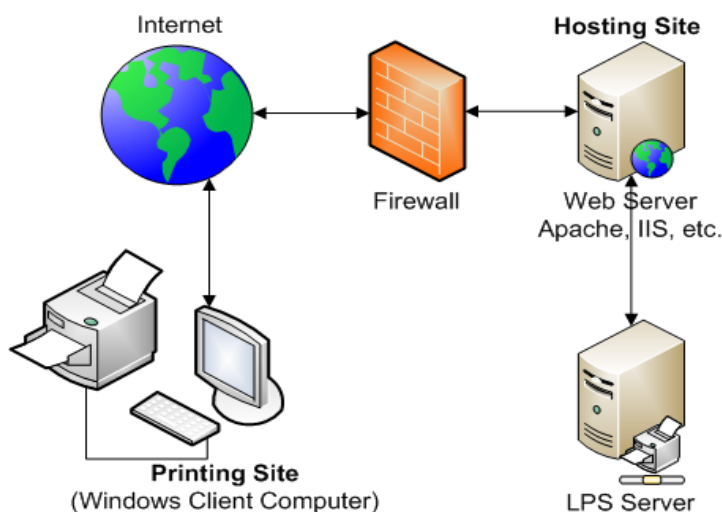


Figure 6-A: The Concept of Internet ActiveX

Internet ActiveX Control

This chapter documents the Loftware Internet ActiveX Control. Like the WebClient and Loftware's other ActiveX Client Controls, iX has a "thin" footprint because it does not require that the Loftware Label Manager subsystem be installed on the same PC as the control. It is called the "Internet ActiveX" because it acts as a client across the Internet to the LPS (Loftware Print Server). Use this control when your application is running in several places, needs to access many printers, and requires a small footprint.

Installation/Use of the Internet ActiveX Control

Pre-Install Checklist

Have you installed the following?

- Loftware Print Server

Note: The Loftware Print Server can be installed on any 2000/2003/XP Professional computer anywhere on your LAN or WAN.

- Web Server
- Servlet Engine
- LPS Web Servlet
- Client-defined Printers

Hint: Before proceeding, install and run the WebClient from a Client PC to ensure that the Loftware Print Server installation is successful .

Installing the Internet ActiveX Control

Install the Internet ActiveX Control to the Client PC one of the following ways:

- A Full install from the CD adds the web x dll.
- A Client install from the LPS Client Install Folder on the CD.
- An Internet ActiveX Control install, either from the CD, or hosted on the Web Server and downloaded by the Client PC.

Using the Internet ActiveX Control

By running the sample program, called Internet ActiveX Sample 1, and examining the code, you will be able to see how each property, method and event of the Internet ActiveX control is used. Sample programs are installed to Loftware Labeling\Sample Programs with the Loftware Print Server Premier Edition.

The sample program can be run from the PC where Loftware is installed or a client PC. To run the program from a client PC, install the Loftware Print Server

WebClient from the LPS Client Install folder of the Software CD. Before opening the sample program:

- Configure the connected printer as CLIENT DEFINED (using Software Label Manager) on the server.
- Design a label format in Software Label Manager. Make sure the label you design prints successfully from Software Label Manager.

Once you complete the previous steps, you can open the Internet Active X Control Sample 1 program from the Software Labeling\Sample Programs folder.

- Open on InternetActiveControlSample1.exe.
- Open the .vbp file to open the program in Visual Basic, if Visual Basic is installed on your computer.

Software's Internet Active X Control Sample 1

Software's iXControl

ATTENTION: Items in RED are required fields in order to connect and print a label through the web server. Items in BLUE are the Methods, Properties or Events used with the associated object.

STEP 1 Enter the following data:

* Enter URL to Web Server * Enter the Port

Enter the User Name Enter the Password

User Name and Password are optional. They are used if security is needed for access to the Internet.

STEP 2 Press "Connect" to connect to the Web Server

Connect CWebX1.Connect (Call Statement)

Disconnect CWebX1.Disconnect

Connection Successful? CWebX1.isConnected

Exit

STEP 3 Press the "Get Printers" button to return a list of configured printers on the LPS

Get Printers CWebX1.PrinterAlias CWebX1.PrinterName CWebX1.PrinterPort CWebX1.PrinterCount

STEP 4 Enter the Printer Number from the list in step 3 and the Printer Port the label will be printed to on this computer. Press the "Set Printer" button.

Printer Number: * 1 * Set Printer CWebX1.SetPrinter (Call Statement)

Printer Port: * COM2 * Press the "Test Print" button to eject a label from the printer to test communication with the printer.

Printer Timeout: * 8 * Test Print CWebX1.TestPrinter (Call Statement)

STEP 5 Enter the label name to print and press "Set Label Name"

LabelName: * [Text Box] * Set Label Name CWebX1.SetLabelName

Trim Leading Spaces CWebX1.TrimLeadingSpaces

STEP 6 Highlight a field and enter data to print for the highlighted field in the "Field Data" text box and press "Set Data".

[Text Box] CWebX1.FieldCount CWebX1.FieldName CWebX1.FieldLength

FieldName: [Text Box]

Field Data: [Text Box]

Click here to queue the present job, then enter more data to print. Append Job CWebX1.AppendJob

* Set Data CWebX1.SetData

STEP 7 Enter the Quantity of labels to print and press "Print Job"

* Quantity Duplicates: [Text Box] [Text Box] * CWebX1.Quantity CWebX1.SetDuplicates

Cancel Print Job CWebX1.ClearAllData CWebX1.ResetJob

* Print Job CWebX1.PrintJob

Follow the steps that are displayed on the form to print a label. The Blue items are the associated properties, methods or events associated with the corresponding object. Items with Red asterisk are required fields.

1. Create a project in a programming tool that can utilize ActiveX Controls.
2. Add the Loftware Internet ActiveX Control to the new project.
3. Connect to the Web Server by invoking the Connect Method.
4. Select a printer by invoking the SetPrinter Method.
5. Choose a label by invoking the SetLabelName Method.
6. Set field data by using the SetData Method.
7. Print the label format by using the PrintJob Method.

Related Information

For instructions on how to install the Loftware Print Server, refer to The Loftware Print Server section of this guide

For information on installing the Web Server, Servlet Engine, and LPS Web Servlet, refer to the Internet Printing section of this guide.

For information on configuring client-defined printers, refer to the Device Connections section of the Loftware Label Manager User's Guide.

Troubleshooting the Internet ActiveX Client Control

Important: It is critical that you trap error events. This is especially true if the LPS server is running in a clustered environment and a failover occurs. Many of the iX methods throw critical errors during the failover transition.

- The control uses a list inside the label format that is generated when the label is saved. Error Event #26503 is thrown if the label format does not contain the list.
- Make sure that you have some labels designed and printers configured on the LPS as CLIENT DEFINED before trying to use the PrintJob Method.
- Make sure you trap errors and display the error string in the ErrorEvent. This saves considerable debug time.
- You may want to add a multi-line list box to your application that you can add a line to for each WarningEvent and InfoEvent. This provides valuable information that helps you get up and running quickly. When you have reached a level of confidence, you do not need to display these events.

Related Information

For more information on iX errors, refer to the Internet ActiveX Error Event section in this guide.

Design Scenario and Distribution

Overview

The iX control can be integrated into the user or integrator's applications running on remote systems. The Software Print Server (LPS) is installed at a central location driving the connected clients through the Software Web Servlet component. Printers that are not normally accessible from the central location can be driven with the iX control.

Design Scenario

Your company has a central site and remote locations not connected via a WAN. This could be as simple as the building next door, a store in the next city, a plant or hub in the next state, or a facility in another country. No matter the scenario, the only [reasonable] connectivity to the site(s) is through the Internet (dial-up, frame, T1, etc.). You have an application that is installed or accessed at the site(s) that has the ability to integrate ActiveX components. The clients that access your application have connectivity to the Internet.

You can include the iX control in your application to allow for printing at each client location. The Software Print Server as well as the Software Web Servlet can be installed at the central location. The iX control only needs to know the address of the central web server, to establish and maintain connectivity. You also need to let the iX control know about the location of the locally accessible printer(s).

The iX control can also be used internally as well. The necessary component is the Software Web Servlet (running on the Web Server). If you are looking for an Intranet solution, the iX control can communicate to the Software Print Server through your internal Web Server. If you already have a WAN and have an intranet application, the iX control can be added to it. Configuration and Operation are the same whether internal or external; it is the location of the Web Server that determines the architecture.

The iX control relies upon the Software Print Server for licensing; you can include the control in your product and distribute it to anyone who requires it. Only when you decide to 'connect' to the LPS will you require a license. This

allows integrators and end users the flexibility to include the functionality at design time.

Distributing the iX Control with your Application

As mentioned above, the iX Control can be distributed to anyone who requires it. Licensing constraints are handled by the LPS that the Control connects to. When building an install program for your application, include the required Software files with it. This way, your setup program takes care of installing the Software control and its associated files, thereby avoiding having to run two setup programs. The following files are needed to use the control. These files are found in the ...Internet ActiveX/redist directory on the Software CD. If your programming language has a setup wizard, it probably picks up the correct files.

WebX.tlb -> winsys folder

WebX.dll (the control) -> winsys folder. This file needs to be registered.

Internet ActiveX Properties

Loftware's ActiveX Properties are as follows:

Duplicates Property

Syntax

```
Object.Duplicates  
Type = short, read/write
```

Description

The Duplicates property is a read/write property that sets the amount of duplicate labels to print. The default for this property is one. Duplicate labels are EXACT copies of the original label. When using this property with a label that has an incrementing or decrementing serial number, this many labels print before incrementing (or decrementing) the number. To create multiple labels with unique serial numbers, use the Quantity Property.

Example

```
'print 2 copies of each serial number 5 times.  
'the total number of labels printed = 10  
CWebX1.SetLabelName "WebSerial.lwl"  
CWebX1.SetData 0, "ABC-123"  
CWebX1.Quantity = 5  
CWebX1.Duplicates = 2  
CWebX1.PrintJob
```

FieldCount Property

Syntax

```
Object.FieldCount  
Type = short, read only
```

Description

The FieldCount property describes the array created when the SetLabelName method is invoked. It is important to understand that this array is the key to having access to all fields in your label in a dynamic fashion. Fields in the array are accessed by their index number in the array or by the field name itself. Your program may not know ahead of time which and how many fields are in the

label. This is why you to iterate through the array with an index. If you do know your field names ahead of time, it is easier to set their data using the 'FieldName' property.

The FieldCount property is a read only property that displays how many variable fields there are in the current label format. This allows you to iterate through the field array that is generated when the 'SetLabelName' method is invoked. There is no default for this property.

Example

```
Dim i As Integer
  Dim fn As String
  Dim fs As Integer
  Dim line As String

For i = 0 To CWebX1.FieldCount - 1
  Err.Clear
  fn = CWebX1.FieldName(i)
  If (Err.Number <> 0) Then
    line = i & " - Error"
  Else
    fs = CWebX1.FieldLength(i)
    line = i & " - " & fn & " Len: " & fs
  End If
```

FieldLength Property

Syntax

```
Object.FieldLength (index) or (FieldName)
Type = short, read only
```

Description

The 'FieldCount', 'FieldName' and 'FieldLength' properties describe the array created when the 'SetLabelName' method is invoked. It is important to understand that this array is the key to having access to all fields in your label in a dynamic fashion. Fields in the array can be accessed by their index number in the array or by the field name itself. Your program may not know ahead of time which and how many fields are in the label. This is why we allow you to iterate through the array with an index. If you do know your field names ahead of time, it is easier to set their data using the 'FieldName' property.

The `FieldLength` property is a read only property that displays the length of a specified field in the current label format. This property can be retrieved by the actual field name or field index number. This property is not filled until the `SetLabelName` method has been invoked.

Note: This property is very useful for validating data before actual printing to assure the number of characters supplied for the field does not exceed its `Field Length`.

There is no default for this property.

Example

```
Dim i As Integer
Dim fn As String
Dim fs As Integer
Dim line As String

For i = 0 To CWebX1.FieldCount - 1
  Err.Clear
  fn = CWebX1.FieldName(i)
  If (Err.Number <> 0) Then
    line = i & " - Error"
  Else
    fs = CWebX1.FieldLength(i)
    line = i & " - " & fn & " Len: " & fs
  End If
```

FieldName Property

Syntax

```
Object.FieldName ( index )
Type = short, read only
```

Description

The 'FieldName' property is created when the 'SetLabelName' method is invoked. It is important to understand that this array is the key to having access to all fields in your label in a dynamic fashion. Fields in the array can be accessed by their index number in the array or by the field name itself. Your program may not know ahead of time which and how many fields are in the label. This is why we allow you to iterate through the array with an index. If you

do know your field names ahead of time, it is easier to set their data using the 'FieldName' property.

The FieldName property is a read only property that displays the name of a specific field in the current label format. This property can only be retrieved by the field index number. This property is not filled until the SetLabelName method has been invoked. There is no default for this property.

Example

```
Dim i As Integer
Dim fn As String
Dim fs As Integer
Dim line As String
For i = 0 To CWebX1.FieldCount - 1
  Err.Clear
  fn = CWebX1.FieldName(i)
  If (Err.Number <> 0) Then
    line = i & " - Error"
  Else
    fs = CWebX1.FieldLength(i)
    line = i & " - " & fn & " Len: " & fs
  End If
```

isConnected Property

Syntax

```
Object.isConnected
Type = boolean, read only
```

Description

The isConnected property checks to see if the currently selected Loftware Print Server is connected. If the LPS is not found, or is not connected, an error is thrown in the ErrorEvent. Make sure that you handle this error in the ErrorEvent, if you do NOT, your program may crash!

Example

```
If (CWebX1.isConnected = True) Then
  MsgBox("LPS is Successfully Connected")
Else
  MsgBox ("Not Connected")
End If
```

JobName Property

Syntax

```
Object.JobName  
Type = string, read/write
```

Description

JobName is a read/write property reflecting a unique identifier for the current job. The default value changes after the PrintJob method is called and follows this naming convention:

ComputerName + "X" + unique instance number + "_" +
YYYYMMDDHHNNS + serial number per second

If you wish to specify your own JobName, it is up to the creator to ensure its uniqueness. It is through this identifier that future job status is returned. The JobName is also reflected in the Status View console that monitors the status and progress of the jobs processed by the LPS. This property is for feedback purposes only. It is not needed to print labels.

Example

```
Dim jobName as string  
' Get the current JobName to store for reference  
jobName = CWebX1.JobName
```

Pages Property

Syntax

```
Object.Pages  
Type = short, read/write
```

Description

The Pages property is a read/write property that sets how many pages of labels are printed. A page of labels is a copy of an entire page of labels created when printing with layouts. The default for this property is one. For more detail on pages of labels, consult the *Loftware Label Manager User's Guide*. Do not use this property if your label does not use a "multi-up" layout.

Example

```
'print 2 identical pages, line break, comment,  
'10 labels/page with 2 labels for each serial #
```

```
CWebX1.SetLabelName "WebSerial.lwl"  
CWebX1.SetData 0, "ABC-123"  
CWebX1.Quantity = 5  
CWebX1.Duplicates = 2  
CWebX1.Pages = 2  
CWebX1.PrintJob
```

PrinterAlias Property

Syntax

```
Object.PrinterAlias (pindex)  
Type = string, read only
```

Description

Table of printer aliases used in conjunction with: PrinterAlias, PrinterCount, PrinterName, PrinterPort, PrinterNumber, that is populated during Connect().

PrinterAlias is used along with PrinterName, PrinterCount, and PrinterPort properties to allow you to prompt your users for the target-configured printer to which they wish to print. Essentially, these properties expose the list of configured printers that was set up in the Loftware label design mode. Printer Alias is a descriptive name for a printer that is assigned when in the printer connection dialog box of label design mode (same dialog box where you specify a port/spooler/IP address).

Example

```
Dim i As Integer
Dim pn As String
Dim pp As String
Dim line As String

For i = 1 To CWebX1.PrinterCount
    pn = CWebX1.PrinterName(i)

    If (Err.Number <> 0) Then
        line = i & " - Not Configured"
    Else
        pp = CWebX1.PrinterPort(i)
        pa = CWebX1.PrinterAlias(i)
        line = i & " - " & "Alias" & " (" & pa & ") " & pn &
" on " & pp
    End If

    List1.AddItem (line)
Next
```

PrinterCount Property**Syntax**

```
Object.PrinterCount
Type = short, read only
```

Description

PrinterCount is used to return the number of printer seats on the license key. Along with PrinterName, PrinterPort, and PrinterAlias properties, it provides information on the configured printers to which users wish to print. These properties expose the list of configured printers that are set up in the Software label design mode as CLIENT DEFINED.

Example

```
Dim i As Integer
Dim pn As String
Dim pp As String
Dim line As String
For i = 1 To CWebX1.PrinterCount

    Err.Clear

    pn = CWebX1.PrinterName(i)
    If (Err.Number <> 0) Then line = i & " - Not
Configured"
    Else
        pp = CWebX1.PrinterPort(i)
        pa = CWebX1.PrinterAlias(i)
        line = i & " - " & "Alias" & " (" & pa & ") " & pn
& " on " & pp
    End If
    List1.AddItem (line)
Next
```

PrinterName Property**Syntax**

```
Object.PrinterName (pindex)
Type = string, read only
```

Description

PrinterName allows you to be able to prompt users for the target-configured printer to which they wish to print. Essentially, this property exposes the list of configured printers set up in the Software design mode.

Example

```

Dim i As Integer
Dim pn As String
Dim pp As String
Dim line As String
For i = 1 To CWebX1.PrinterCount
    line = ""
    Err.Clear
    pn = CWebX1.PrinterName(i)
    If (Err.Number <> 0) Then
        line = i & " - Not Configured"
    Else
        pp = CWebX1.PrinterPort(i)
        pa = CWebX1.PrinterAlias(i)
        line = i & " - " & "Alias" & " (" & pa & ") " & pn
        & " on " & pp
    End If
    List1.AddItem (line)
Next

```

PrinterNumber Property**Syntax**

```

Object.PrinterNumber
Type = string, read/write

```

Description

Set by application to indicate the printer number for the print job.

Used in conjunction with: PrinterAlias, PrinterCount, PrinterName, PrinterPort, PrinterNumber.

See the "Printer Lists" section.

The PrinterNumber property is a read/write property that sets which configured LPS printer to print to. The default for this property is configured Printer #1.

Using the PrinterName or PrinterAlias properties, you could prompt your user with a better description of the target printer. You could then resolve their choice into a number, which ultimately must be used in code to specify the printer.

Example

```

Dim P As Integer
P = txtPrinterNum.Text
'Set the printer number, printer port and the

```

```
printer timeout value
    Call CWebX1.SetPrinter(p, txtPrinterPort.Text,
txtPrinterTimeout.Text)
    CWebX1.printerNumber = p
```

PrinterPort Property

Syntax

```
Object.PrinterPort (pindex)
Type = string, read only
```

Description

PrinterPort allows you to be able to prompt your users for the target-configured printer to which they wish to print. They are listed as “Not Configured” or as “CLIENT DEFINED”.

Example

```
Dim i As Integer
Dim pn As String
Dim pp As String
Dim line As String
For i = 1 To CWebX1.PrinterCount
    Err.Clear
    pn = CWebX1.PrinterName(i)
    If (Err.Number <> 0) Then
        line = i & " - Not Configured"
    Else
        pp = CWebX1.PrinterPort(i)
        pa = CWebX1.PrinterAlias(i)
        line = i & " - " & "Alias" & " (" & pa & ") " & pn
        & " on " & pp
    End If
    List1.AddItem (line)
Next
```

PrinterTimeout Property

Syntax

```
Object.PrinterTimeout (pindex)
Type = string, read/write
```

Description

The PrinterTimeout reflects the length of time in seconds that the LPS waits for response from the printer before displaying a timeout message. The default is 8 seconds, which may not be enough time if there are large graphics being sent to the printer that take longer than 8 seconds to load.

This property **MUST** be set in order to obtain the printer information (PrinterCount, PrinterName, PrinterPort, or PrinterAlias).

Example

```
Dim p as Integer
p = txtPrinterNum.Text
'Set the printer number, printer port and the printer
timeout value
Call CWebX1.SetPrinter(p, txtPrinterPort, Text,
txtPrinter.Text)
```

Quantity Property

Syntax

```
Object.Quantity
Type = short, read/write
```

Description

The Quantity property is a read/write property that sets the amount of labels to print. The default for this property is one. The Quantity of labels is used to increment serial numbers and other incrementing fields.

When using this property with a label that has an incrementing or decrementing serial number, it is important to note that changing this property **DOES NOT** create duplicate serial numbers. To create multiple copies of labels with identical serial numbers use the Duplicates Property.

Example

```
'This example ends up producing 10 labels
'with 2 duplicates of any incrementing or decrementing
number
CWebX1.SetLabelName "Label1.lwl"
CWebX1.SetData 0, "ABC-123"
CWebX1.Quantity = 5
CWebX1.Duplicates = 2
CWebX1.PrintJob
```

TrimLeadingSpaces Property

Syntax

```
Object.TrimLeadingSpaces  
Type = Boolean, read/write
```

Description

The TrimLeadingSpaces property trims the leading spaces from data before it printed. If your program has control of the data, you should do any trimming or data manipulation before using the "SetData" Method. If you are going to use this property, it must be set before using the SetData Method. The default for this property is FALSE.

Example

```
'This example ends up producing 10 labels  
'The spaces before "ABC" are removed by the system  
CWebX1.SetLabelName "Label1.lw1"  
CWebX1.TrimLeadingSpaces=TRUE  
CWebX1.SetData 0, " ABC-123"  
CWebX1.PrintJob
```

WebAddress Property

Syntax

```
Object.WebAddress  
Type = string
```

Description

The web URL for the web server that connects to the LPS server. Where WebAddress is a character string containing the TCP/IP address in the standard format, or Hostname such as "www.loftwarelabeling.com".

Example

```
CWebX1.WebAddress = "172.15.0.99"
```

WebPort Property

Syntax

```
object.WebPort  
type = string
```

Description

The WebPort Property sets the Web Port to be utilized in the transmission of the data. This is used if a Port other than the default (80) is used.

Example

```
CWebX1.WebPort = "8080"
```

WebUserName Property**Syntax**

```
object.WebUserName
```

Description

(OPTIONAL) Username for security (if required for access to the Web Server)

Default: null

Example

```
CWebX1.WebUserName = "UserOne"
```

Internet ActiveX Methods

This section describes Loftware's ActiveX Methods.

AppendJob Method

Syntax

```
Object.AppendJob
```

Description

AppendJob is used to queue up the current label that has been designated via the SetLabelName method calls, as well as Quantity, Duplicates, Pages, and/or PrinterNumber properties. AppendJob is used for batching label requests together for one server request, instead of many separate jobs.

Example

```
Private Sub cmdAppendJob_Click()  
    Adds the current job to a queue  
    CWebX1.AppendJob  
End Sub
```

ClearAllData Method

Syntax

```
Object.ClearAllData
```

Description

ClearAllData is used to clear all the data members for the current label, ensuring there are no 'sticky' data values being sent for the next job.

Example

```
Private Sub cmdCancel_Click()  
    'Clears data queued  
    CWebX1.ClearAllData  
    'Removes all queued jobs  
    CWebX1.ResetJob  
End Sub
```

Connect Method

Syntax

```
Object.Connect
```

Description

Connect the iX control to the Web Server and JSP. If the JSP is successful in establishing a session with LPS, Connect returns TRUE.

Otherwise, Connect returns FALSE and an ErrorEvent is generated.

Note: THIS METHOD SHOULD BE CALLED BEFORE ANY MANIPULATION OF THE CONTROL.

Example

```
Call CWebX1.Connect (txtURL.Text, txtPort.Text,  
txtUserName.Text, txtPassword.Text)
```

Disconnect Method

Syntax

```
Object.Disconnect
```

Description

This method disconnects the iX Control from the Web and LPS connection. If there are no errors, Disconnect returns TRUE. If there is an error, Disconnect returns FALSE.

Example

```
CWebX1.Disconnect
```

PrintJob Method

Syntax

```
Object.PrintJob
```

Description

PrintJob is used to submit the current job to the LPS via the internet. iX client control creates a Pass file with all queued jobs and sends it to the LPS over the web. The current data is supplied via the SetData method calls, as well as

Quantity, Duplicates, Pages, and/or PrinterNumber properties. The job is then created as a print stream by the LPS, sent back to the client control, and printed.

Example

```
CWebXl.Quantity = txtQuantity.Text
CWebXl.Duplicates = txtDuplicates.Text
CWebXl.PrintJob
```

ResetJob Method

Syntax

```
Object.ResetJob
```

Description

ResetJob is used to clear the entire job, including any labels that have been appended to the batch with the AppendJob method. This is useful for instances where the end user has changed their mind and wishes to cancel the job.

Example

```
Private Sub cmdCancel_Click()
'Clears data queued
CWebXl.ClearAllData
'Removes all queued jobs
CWebXl.ResetJob
End Sub
```

SetData Method

Syntax

```
Object.SetData
```

Description

The 'FieldCount', 'FieldName', 'FieldData', and 'FieldLength' properties describe the array that is created when the 'SetLabelName' method is invoked. SetData accesses this array and populates the field specified by either index number or field name with the data specified. Your program may not know ahead of time which and how many fields are in the label. This is why we allow you to iterate through the array with an index. If you do know your field names ahead of time, it is easier to set their data using the 'FieldName' property.

Note: You must do a 'SetLabelName' before you can set data using 'SetData'. You do not have to redo a 'SetLabelName' after doing a 'PrintJob', unless you are changing to a different label format.

Example1

```
Call CWebX1.SetData(txtFieldName.Text, txtData.Text)
```

Example 2

```
'setting data by iterating through field count instead  
of using field name  
'this example sets every field to "sample data"  
CWebX1.SetLabelName "ibm.LWL"  
For I = 0 to CWebX1.FieldCount - 1  
    CWebX1.SetData I, "sample data"  
Next I  
CWebX1.PrintJob
```

SetLabelName Method

Syntax

```
Object.SetLabelName
```

Description

The SetLabelName method accesses the label across the web and retrieves the field information for the specified label format. This method also creates an array to hold the variable data for every variable field on the current label format. This array is accessed by index number or field name. If you do not know the field names ahead of time, you must use an index into the array. See 'FieldCount' and 'SetData' for more information on this.

Note: Because the array is contained within the control, it is Private. The only way to get information to and from the array is through the control's properties and methods. The Array that is created has the following structure:

FieldName() as a String

FieldLength() as a short

FieldData() as a String

The first index number of the array is zero.

Example

```
CWebX1.SetLabelName "AIAG.LWL"
CWebX1.SetData "PARTNUMBER", "A100"
CWebX1.PrintJob
```

SetPrinter Method**Syntax**

```
Object.SetPrinter
```

Description

The SetPrinter Method sets the following parameters for the local printer:

- Index into local tables (the printer number)
- The local port (PrinterPort)
- A timeout value (PrinterTimeout)
- Sets the corresponding properties

Example

```
CWebX1.SetPrinter
Dim p As Integer
p = txtPrinterNum.Text
Call CWebX1.SetPrinter(p, txtPrinterPort.Text,
txtPrinterTimeout.Text)
CWebX1.printerNumber = p
```

Test Connection Method**Syntax**

```
Object.TestConnection
Type = Boolean
```

Description

This method tests the URL/Port connection to servlet. Similar to the "Test" button on the WebClient Address/URL dialog box.

Example

```
Dim nPort as integer or short
Err.Clear
nPort = CInt(txtPort.Text)
Call CWebX1.TestConnection(txtURL.Text, nPort,
txtUserName.Text, txtPassword.Text)
If (Err.Number <> 0) Then
    MsgBox ("Test Connection Failed!")
Else
    MsgBox ("Test Connection OK!")
End If
```

TestPrinter Method

Syntax

```
Object.TestPrinter
```

Description

This method tests the local printer denoted by "PrinterNumber" by sending a formfeed to the printer. This is useful to see if your locally configured client-defined printer is configured / connected correctly. If communication is successful, a label ejects with the word TEST printed on the label, and the message "Check printer for the Label" is displayed. If there is no communication with the printer, a "Test Failed" message box is displayed.

Example

```
Dim p As Integer
    p = txtPrinterNum.Text
CWebX1.TestPrinter (p)
```

Internet ActiveX Events

This section describes Loftware's iX Events.

AfterPrint Event

Syntax

```
Object_AfterPrint()
```

Description

The after print event is an event that is fired just after closing the port once the print stream has been sent, and is used for specialized printing needs.

Example

```
CWebX1_AfterPrint
```

BeforePrint Event

Syntax

```
Object_BeforePrint()
```

Description

The before print event is an event that is fired just before opening the port to send the print stream, and is used for specialized printing needs.

Example

```
CWebX1_BeforePrint
```

ErrorEvent Event

Syntax

```
Object_ErrorEvent()
```

Description

Passes error messages back to the container during various operations. ErrorEvent items are critical and indicate that an error is generated (caught by error handling). An operation might send multiple error events, but the first occurrence of the ErrorEvent is the error that is passed back to the container to the error handling routines. See the ErrorEventIDs table for a breakdown of all

warning events that are generated. Make sure that your program takes appropriate action after receiving one of these events. The safest bet when it comes to the error event is to end the program.

Note: It is critical that you trap the error event. This is especially true if the LPS server is running in a clustered environment and a failover occurs. Many of the ClientX methods throw critical errors during the failover transition. The following error trap handles this scenario:

Example

```
'display a message box showing the error number and
string
Private Sub CWebX1_ErrorEvent(ByVal errorID As Long,
ByVal errorString As String)
    Debug.Print errorString
End Sub
```

ErrorEvent Ids and Messages

(Also includes error returned from the control)

ID	Message	Explanation
25503	Critical Failure of job # jobNum	A critical failure of a job has been reported
25504	Error on printer prnternumber	The server is reporting a printer error on the listed printer. (Not currently used in the control)
25505	Printer Name: Not Found	The printer name was not located in the LPS printer configuration file
25506	Printer Port: Not Found	The printer port was not located in the LPS printer configuration file
25507	Printer ID: Not Found	The printer id was not located in the LPS printer configuration file .
25512	Failed to obtain the Computer Name!	The computer name is required to generate the unique jobname as well as unique filenames.
26500	Unable to open file.	Unable to open the label file
26501	Unable to open file 'file'.	Unable to open the label file
26502	Unable to change file mode for 'file'	Internal error when opening and parsing the label file. (Try again)
26503	Tab marker information not located in file 'file'	The requested label file has been saved under a previous version. Open and save the label file to the newest version
26504	Label not set	An attempt to access a field or print a job with having called

ID	Message	Explanation
		SetLabelName() first.
26505	Not a valid index number	An invalid index number. (<0 or greater than the count)
26506	Fieldname 'name' not found	A call to SetData or Fieldlength with the invalid fieldname 'name'.
26509	Not a Valid Printer Number	A negative printer number was passed
26510	Printer alias 'alias' not found	Printer alias was not found.
26514	Value cannot be negative.	Quantity, Duplicates, Pages, or PrinterNumber was set to a negative number.
26515	Could not determine the fieldname/position	Call to SetData with a zero-length string for a fieldname.
27501	Not connected to Server	Connect method must be called before calling other methods
27502	Error Retrieving Data from LPS	A communications level error occurred while transacting with the Print Server. Try the operation again, or disconnect
27503	Error processing request from LPS	An error occurred while processing a request on LPS. The textual message is included.
27504	Error Printer has not been locally configured	An attempt to print to a printer without calling SetPrint0 first.
27505	Error Control not properly initialized	An attempt to print a job without a valid connection.
27506	Error Print Job Totally Failed	The entire print request failed
27507	Failed to initialize the printer	Failed to connect to the printer, check configuration to ensure proper connection
27508	Failed to generate the print request	Internal error attempting to send the print request to LPS. Disconnect, reconnect and try again.
27509	Invalid UserName/Password	An attempt to connect to the Web Server failed with an invalid UserName and/or Password.
27510	The connection with LPS has been dropped! Please restart.	The connection has been severed. LPS, Web Server or the Internet connection may have been shut down. Disconnect, reconnect and try again.
27511	Received error xx from Web Server/Servlet	An error from the Loftware Web Servlet has been reported. Please note the number, disconnect, reconnect, and then try again. If this persists, contact Loftware Technical Support for further assistance.

InfoEvent Event

Syntax

```
Object_InfoEvent()
```

Description

The InfoEvent Event passes informational messages back to the container during various operations. InfoEvent items are used mainly as checkpoints and can be useful if displayed to the end user during initialization, etc. See the InfoEvent ID table for a breakdown of all info events that may be generated. This is a useful event during the development/debug process of your program. Once your program is fully debugged, you do not have to pay much attention to this event.

Example

```
Private Sub CWebX1_InfoEvent(ByVal infoID As Long,  
ByVal infoString As String)  
    Debug.Print infoString  
End Sub
```

InfoEvent IDs and Messages

ID	Message	Explanation
25005	PasExt: ext	Displays the extension used for Pass files.
25006	Name: name	Displays the ServerName.
25007	Alias: alias	Displays the ServerAlias.
25008	Gathering Printer Information	Checkpoint before obtaining the printer info.
25009	Printer number	Displays the PrinterNumber
25010	Printer name:	Displays the PrinterName
25011	Printer Port: port	Displays the PrinterPort
25012	Printer ID: number	Displays the printer ID
25013	Printer Alias: alias	Displays the PrinterAlias
25018	Getting information for entry: section	Lists the current section in the ini file being processed.
25019	Verified Path Exists	Path existence test OK.
25020	Verified Write Access	Path write test OK
25021	Verified Existence of type files	Files with extension of type were found in directory
25022	Verified Read Access for one file: file	Verified the ability to open a file for read (filename is displayed).
25023	Verified Read Access	Verified read access to a directory
25025	Verified File Exists	A file has been checked for existence
27001	Printed Job Number xx to printer on port	Job Printed Info Message

OtherEvent Event

Syntax

```
Object.OtherEvent()
```

Description

Other events are not currently used, but they contain information that does not fit into the info, warning, or error categories.

Example

```
Private Sub CWebX1_OtherEvent(ByVal otherID As Long,
ByVal otherString As String)
    Debug.Print otherString
End Sub
' Log these messages to an info multi-line textbox
txtOther.Text = otherString
End sub
```

WarningEvent Event

Syntax

```
Object.WarningEvent()
```

Description

Passes warning messages back to the container during various operations. WarningEvent items are used to flag a missing item, etc., which is not detrimental to the operation, but is out of the ordinary. See the WarningEventID table for a breakdown of all warningevents that may be generated.

Example

```
'display a message box showing the error number and
string
Private Sub CWebX1_WarningEvent(ByVal warningID As
Long, ByVal warningString As String)

MsgBox("Warning # " & warningID & " has occurred.
Message = " & warningString)
End sub
```

WarningEvent IDs and Messages

ID	Message	Explanation
25250	Alias Not Found!	Server Alias was not located (optional entry) in the llmwcInt.ini file.
25254	PrinterAlias: Not Found	There was no Printer Alias defined for the particular printer.
25256	Could not locate any 'type' files	No files of type "type" were located when checking the path.
25257	Unable to test Read Access!	Since there were no files located, read access on files cannot be checked.
25258	Failed to obtain the Username of the currently logged in user	The username of the current logged in user is unable to be determined. (Used for job tracking, on the back end).
27251	There were errors processing Job!	There were errors encountered by LPS while processing the job/request.

Internet ActiveX Reference Table

Name	Read	Write	Type	Default	Comment
AfterPrint Event	N/A	N/A	Event	N/A	Event fired just after closing port
BeforePrint Event	N/A	N/A	Event	N/A	Event fired just after opening port
AppendJob	N/A	N/A	Method	N/A	Batches label requests together.
CancelOperation	N/A	N/A	Property	N/A	Cancels data transmission
ClearAllData	N/A	N/A	Method	N/A	Clears all the data members for the current label
Connect	N/A	N/A	Method	N/A	Connects to the Web Server
Disconnect	N/A	N/A	Method	N/A	Disconnects from the Web Server
Duplicates	X	X	Property	1	Sets the amount of duplicate labels to print.
ErrorEvent	N/A	N/A	Event	N/A	Passes error messages back to the container.
Field Count	X		Property	N/A	Displays how many variable fields there are in the current label format
FieldLength	X	N/A	Property	N/A	Displays the length of a specified field in the current label format
FieldName	X		Property	N/A	Displays the name of a specific field in the current label format.
InfoEvent	N/A	N/A	Event	N/A	Passes informational messages back to the container.
isConnected	N/A	N/A	Property	N/A	Checks to see if the program is connected to the Web Server
JobName	X	X	Property	Contains computer name	Reflects a unique identifier for the current job.
OtherEvent	N/A	N/A	Event	N/A	Contains information that does not fit into the info, warning or error categories
Pages	X	X	Property	1	Sets how many pages of labels are printed.
PrinterAlias	X		Property	N/A	The descriptive name for an assigned printer.
Printer Count	X		Property	N/A	The number of configured printers.
PrintJob	N/A	N/A	Method	N/A	Used to submit the current job to the LPS.
PrinterName	X		Property	N/A	Name of configured printer.
PrinterNumber	X	X	Property	1	Corresponds to configured Printers
PrinterPort	X		Property	N/A	List of configured printer ports

Name	Read	Write	Type	Default	Comment
PrinterTimeout	X		Property	N/A	Sets the time in seconds before Printer Timeout message is displayed.
Quantity	X	X	Property	1	Sets the amount of labels to print.
ResetJob	N/A	N/A	Method	N/A	Clears the entire job, including appended labels.
SetData	N/A	N/A	Method	N/A	Populates the field by index number or field name.
SetLabelName	N/A	N/A	Method	N/A	Retrieves field information for the specified label format.
SetPrinter	N/A	N/A	Method	N/A	Tells the control which printer to use.
TestConnection	N/A		Property	0	Tests the connection to the Web Server
TestPrinter	N/A	N/A	Method		Sends a test to the printer to see if properly configured
TrimLeading Spaces			Property		Trims spaces off of label data
WarningEvent	N/A	N/A	Event	N/A	Passes warning messages back to the container during various operations.
WebAddress	X		Property		The Web IP Address or URL for the web server that connects to the LPS server.
WebPort	X		Property	80	Sets the WebPort number if you are not using the default
WebUserName			Property	Null	Username for security (Used if required for access to the Server)