

Loftware Print Server

User's Guide

Copyright© 2004 – 2006. All rights reserved. Revision 23-8.x

Loftware, Loftware Label Manager, LLM, LLM-WIN, LLM-WIN32, LPS Premier Edition, LPS Internet Edition, Loftware Print Server, Loftware Connector, LPS Print Server Edition, and LLM-WIN Print Module are registered trademarks of Loftware, Inc. Other brands and product names are trademarks, registered trademarks, or registered trademarks of their respective holders.



LOFTWARE, INC.

Global Marking Solutions for Enterprise Applications™

*Barcode Label Printing Software and Integration Solutions for PCs, Networks,
Mid-Range and Mainframe Computers*

166 Corporate Drive, Portsmouth, NH 03801 U.S.A.

Tel: (603) 766-3630 Fax: (603) 766-3631

Internet Website: www.loftware.com



Contents

SOFTWARE SUPPORT AND SERVICES.....	I-1
About Software	I-1
User Manuals	I-2
Contacting Software.....	I-2
Technical Support	I-3
Professional Services	I-4
CHAPTER 1 THE SOFTWARE PRINT SERVER.....	1-1
Overview of the LPS	1-1
Understanding the LPS Architecture.....	1-1
Performance Considerations	1-5
Installing the LPS	1-7
System Requirements for the LPS.....	1-7
System Analyst Questions.....	1-8
Pre-Install System Checks.....	1-8
Installing and Verifying	1-10
Configuration of the LPS	1-12
General Tab.....	1-12
Logging Tab.....	1-16
Notification and Status Reporting Tabs.....	1-18
Housekeeping Tab.....	1-18
LPS Modes	1-20
Service Mode	1-20
Interactive Mode	1-21
Print Request Data Structures	1-22
Providing Data for the General Case.....	1-29
LPS Interfaces	1-33
The File Interface	1-34
ActiveX / .NET Interface	1-34
Connector Interface	1-35
Direct Socket Interface	1-35
TCP/IP Socket Interface.....	1-36
Hints and Troubleshooting	1-42
WDPING and WDLOG Diagnostic Utilities	1-43
Reading the Application Event Log	1-46
CHAPTER 2 THIN CLIENT MODULES.....	2-1
Thin Client Overview.....	2-1
Installing and Starting Client Applications	2-2

Notification Agent.....	2-2
Configuring the Notification Agent	2-3
Starting the Agent	2-7
Testing and Troubleshooting the Notification Agent.....	2-9
On-Demand Print Client	2-13
Preparing to Use the On-Demand Print Client.....	2-14
The llmwclnt.ini File.....	2-15
Operation of the On-Demand Client	2-16
Creating an Icon to Run the Thin On-Demand Print Client.....	2-19
Troubleshooting the Thin On-Demand Print Client.....	2-20
Status Client Application	2-21
Understanding the Screen Layout	2-22
Status Client Examples	2-24
Status Menu Choices	2-26
Using the 'Launch Browser' Feature	2-29
LPSSend Client.....	2-30
Palm OS SDK	2-30
Running the Sample Program	2-31
CHAPTER 3 INTERNET PRINTING	3-1
Internet Printing Overview.....	3-1
Internet Printing Considerations	3-1
The WebClient.....	3-2
Choosing the WebClient or On-Demand Print Client.....	3-4
Components of the WebClient.....	3-6
Installing the WebClient Components	3-8
Using the WebClient.....	3-18
Additional WebClient Information	3-20
Internet Data Push and the Web Listener.....	3-22
Installing the Web Push Components	3-24
Configuring the Web Push Components.....	3-24
Client-side Connections.....	3-27
Client-side Printer Configurations	3-31
Using Web Listener as a Service	3-35
CHAPTER 4 LPS CLUSTERING	4-1
What is Clustering?.....	4-1
Definition of Terms	4-2
Software Print Server (LPS) Failover Support.....	4-4
Shared Files and Central Storage	4-5
The Mirrored Registry	4-5
Installing the Software Print Server on a Cluster.....	4-6
Upgrading, Repairing, and Uninstalling	4-18
Remote Administration of a Windows 2000 Cluster Server	4-18
Client Connections to the LPS Virtual Server.....	4-19
Troubleshooting Software Print Server Clustering.....	4-20

CHAPTER 5 ACTIVEX CLIENT CONTROL.....	5-1
ActiveX Client Control Overview.....	5-1
Installation and Use of ActiveX Client Control	5-2
Design Scenario and Distribution.....	5-3
ActiveX Client Control Properties	5-5
ClientIniPath Property.....	5-5
DropDirectory Property.....	5-7
Duplicates Property	5-7
FieldCount Property	5-8
FieldLength Property.....	5-9
FieldName Property	5-10
isRunning Property.....	5-12
JobName Property.....	5-12
LabelsPath Property	5-13
LayoutPath Property.....	5-13
Pages Property.....	5-14
PrinterAlias Property.....	5-14
PrinterCount Property	5-15
PrinterName Property.....	5-16
PrinterPort Property.....	5-17
PrinterNumber Property	5-18
PrinterPath Property	5-18
PasExt Property.....	5-19
Quantity Property	5-19
ServerCount Property.....	5-20
ServerNumber Property.....	5-21
ServerName Property	5-21
ServerAlias Property	5-22
TrimLeadingSpaces Property.....	5-23
ActiveX Methods	5-23
AppendJob Method	5-23
ClearData Method	5-24
Login Method.....	5-25
PrintJob Method.....	5-25
ResetJob Method.....	5-26
SetData Method.....	5-26
SetLabelName Method.....	5-28
SetServer Method.....	5-29
ActiveX Events	5-30
ErrorEvent Event.....	5-30
InfoEvent Event	5-33
WarningEvent Event	5-34
OtherEvent Event	5-35
ActiveX Client Control Reference Table	5-36
 CHAPTER 6 INTERNET ACTIVEX CONTROL.....	 6-1
Internet ActiveX Overview	6-1
Installation/Use of the Internet ActiveX Control	6-2
Design Scenario and Distribution.....	6-3

Internet ActiveX Properties.....	6-5
Duplicates Property.....	6-5
FieldCount Property.....	6-5
FieldLength Property.....	6-6
FieldName Property.....	6-7
isConnected Property.....	6-8
JobName Property.....	6-8
Pages Property.....	6-9
PrinterAlias Property.....	6-10
PrinterCount Property.....	6-11
PrinterName Property.....	6-11
PrinterNumber Property.....	6-12
PrinterPort Property.....	6-13
PrinterTimeout Property.....	6-14
Quantity Property.....	6-14
TrimLeadingSpaces Property.....	6-15
WebAddress Property.....	6-15
WebPort Property.....	6-15
WebUserName Property.....	6-16
Internet ActiveX Methods.....	6-16
AppendJob Method.....	6-16
ClearAllData Method.....	6-17
Connect Method.....	6-17
Disconnect Method.....	6-18
PrintJob Method.....	6-18
ResetJob Method.....	6-18
SetData Method.....	6-19
SetLabelName Method.....	6-20
SetPrinter Method.....	6-20
Test Connection Method.....	6-21
TestPrinter Method.....	6-21
Internet ActiveX Events.....	6-22
AfterPrint Event.....	6-22
BeforePrint Event.....	6-22
ErrorEvent Event.....	6-22
InfoEvent Event.....	6-24
OtherEvent Event.....	6-25
WarningEvent Event.....	6-26
Internet ActiveX Reference Table.....	6-27

CHAPTER 7 SOFTWARE .NET CONTROL..... 7-1

Overview of Software .NET Control.....	7-1
Installation and Use of the Software .NET Control.....	7-2
Installing the Software .Net Control.....	7-3
Using the Software .NET Control.....	7-4
Distributing the Software .NET Control.....	7-5
Using the .NET Control in a Web Application.....	7-5

Loftware .NET Control Methods	7-6
AppendJob	7-6
ClearData	7-6
GetFieldIndex	7-7
GetFieldMaxLength	7-7
GetFieldName	7-8
GetLabel	7-9
GetLabelList	7-10
GetLabelListExt	7-11
GetPrinterByNumber	7-12
GetPrinters	7-13
KillJobs	7-14
Login	7-14
Logout	7-15
PrintJob	7-15
SetData	7-18
Loftware .NET Control Properties	7-19
CurrentLabel	7-19
CurrentServer	7-19
Duplicates	7-20
FieldCount	7-20
JobName	7-21
LabelDtd	7-22
LastErrorMessage	7-22
LoggedIn	7-23
Pages	7-23
PrinterName	7-24
PrinterNumber	7-24
Quantity	7-25
Tray	7-26
Loftware .NET Control Events	7-26
ErrorMessage Event	7-26
WarningMessage Event	7-27
InfoMessage Event	7-27
More Error Information and Return Codes	7-27
Reference Table for the Loftware .NET Control	7-28

INDEX



Loftware Support and Services

About Loftware

Since 1986, Loftware has been providing world-class bar code labeling systems to virtually every industry. From our earlier stand-alone label design and printer workstation products to our modern “Server and Web Centric” based client/server/Internet technologies, Loftware continues its industry leadership position in developing the most advanced systems tools and labeling solutions available on the market today.

Open System Technology that Protects Your Labeling System Investment

Loftware’s printing software platforms are designed for maximum flexibility and maintainability. As labeling, computer and business requirements change, so can we. This not only ensures that your initial investment is protected but guarantees that future reinvestments will not be necessary. Loftware’s platforms also allow for incredible add-on growth, extendibility and scalability. Add and mix different types of printers, print from “thin” network and Internet client printing workstations or roam and print using mobile wireless devices. Print in “stand-alone” mode or integrate printing directly with enterprise business applications such as SAP, Oracle, Baan, JD Edwards operating across LANS, WANS, and the Internet. Because our technology is application independent, any business application can take full advantage of our printing rich platforms without the need for writing extensive custom-built applications or communication interfaces.

With little effort, ERP and WMS systems as well as any other application requiring bar code-printing output can take advantage of Loftware’s state of the art printing platforms.

Early in 2003, Loftware, Inc. made available the Oracle® Connector, which is a new connectivity module bridging Loftware’s current enterprise bar code printing technology to Oracle® applications 8i and above. In mid-2003, Loftware released the first International Edition, allowing users to run Loftware applications in English, French, German, or Spanish.

Today, Loftware is once again ahead in the area of encoding RFID smart labels and tags. Extending its RFID capabilities, Loftware has released its RFID Reader Module (RRM), an optional component of the Loftware Print Server, that allows reading and writing of passive RFID tags with different brands and models of RFID readers. This fast-growing technology is becoming a "must-have" requirement for many suppliers, food vendors, healthcare facilities, and others.

Industry Experience

Loftware is an engineering and support based company, and has highly trained professionals with the experience you need to understand the essential elements of implementing a successful bar code printing system. We have the knowledge in what works and what does not work in real world situations; so please call and speak to our systems analysts and technicians before making a decision!

Loftware's Mottos

"If you CAN'T print labels, you CAN'T ship products!"

"Bar code labels are not just labels, but actual information carriers of your company's data."

"Global Supply Chains start with the label!"

"Never underestimate the complexity or cost associated with implementing a flexible, maintainable and scalable enterprise labeling solution."

User Manuals

This Loftware Print Server (LPS) User's Guide offers advanced information for those seeking an understanding of our advanced enterprise offerings. It is separate from the Loftware Label Manager User's Guide to give those who are already knowledgeable about label-printing a head start in implementing a serious printing solution using advanced technologies. Included in this guide is information about:

- The Loftware Print Server
- Thin Clients
- Internet Printing
- LPS Clustering and redundant systems
- Client, Internet, and .NET Control

The Loftware applications listed above may be run in English, French, German, or Spanish. Our "Quick Start User's Guide" is available in these 4 languages. See Chapter 1 in this guide for instructions.

For information on barcode and RFID label compliance, refer to the 'Loftware Label Manager' manual.

The Loftware RRM Control is described in a separate guide.

Contacting Loftware

Sales and Customer Service

The Loftware's Sales and/or Customer Service departments are available for product information, quotes, and placing orders between 8:00 AM and 5:00 PM Eastern Time. E-mail and Faxes can also be received 24-hours a day.

Phone: 603-766-3630

Fax: 603-766-3631

Email: sales@loftware.com or customerservice@loftware.com

Technical Support

Please refer to the next section entitled “Technical Support.”

Traditional Mail

If you need to contact us via traditional mail, FEDEX, UPS, and/or other mail service carriers, please use Loftware’s shipping address listed below:

Loftware, Inc.
166 Corporate Drive
Portsmouth NH 03801
U.S.A.

Technical Support

Software licenses purchased directly from Loftware include the first year of Technical Support. This initial 12-month support period starts on the day the product is shipped and invoiced from Loftware’s factory. When needed, support recipients during this period are eligible to receive unlimited telephone support, access to software upgrades and enhancements and speak with our Systems Analysts.

Premium Annual Support Contract

To ensure uninterrupted telephone support as well as access to the latest software upgrades and enhancements, make sure all your software licenses remain under a Loftware Support Contract. After your first year of ownership, you will be sent a notice to renew your support contract. Please refer to Loftware’s Internet Website for additional information on this very important topic, or if you prefer, call Loftware’s Customer Service Department for more information.

During the one-year Support Contract period, Contract Subscribers have access to the following services:

1. Unlimited Technical Support Incidents
2. Access to Loftware’s Professional Services Group
3. Automatically eligible to download software upgrades and service packs from our website
4. Automatic e-mail notification when new versions of software become available
5. When necessary, access to senior Loftware technical support staff, via phone and e-mail
6. Guaranteed software license replacement for accidentally damaged or malfunctioning hardware keys

Before Calling Support

Loftware has highly trained technicians available to help you with your labeling system. Technical support calls ***are not*** accepted until ***all*** of the following Technical Support requirements are met:

1. Your product is registered. If you have not registered your software, you may do so at <http://loftware.com> or via fax by using the form included with your software.
2. There is a Support Contract in place that covers the specific license in question.
3. You have checked the user's guide(s) for your answer. If you do not have the User's Guides, both of the guides or various chapters of each can be downloaded in PDF format from our web site, or read on-line. User manuals are also on the Loftware CD.
4. You have checked the Loftware's Knowledge Base articles on our website. Hundreds of frequently asked questions and typical problems are documented there in easy to read articles.
5. If you suspect that your problem is hardware related, try to first determine if it is a problem with your PC, Network, or printer and contact the appropriate company. Loftware does not sell or service any hardware products.
6. Have your serial number and version number of the product you are using ready. These numbers can be obtained by accessing the Help>About menu of the label design mode.
7. Think about how you are going to efficiently explain the problem prior to speaking with a technician. The better the description, the quicker the solution and/or resolution to your problem.
8. If this is a follow up call to a previous incident, please have the incident number ready.

Phone: 603-766-3630

Fax: 603-766-3631

E-mail: support@loftware.com

Professional Services

Systems Analysts

Loftware's Professional Services Group offers "*without charge*" pre-sales printing systems analysis and telephone consulting services to all companies interested deploying a Loftware printing solution. If you are an MIS director, IT Manager or Systems Programmer and have advanced questions on how to best proceed with the design, integration and implementation of a labeling system, we encourage you to contact one of Loftware's PSG Systems Analyst consultants.

The System Analyst is a special added-value service we have developed to help ensure maximum success in integrating our applications with yours. Our Analysts have the real world experience you need to make the right choices early on in your project.

Remember, setting up a labeling system is not an easy task even for the very experienced professional. Assistance 'up front' from experts in the field goes a long way in reducing project times and costs.

PSG On-Site

PSG's Systems Consultants and implementation experts are also available after the sale, to come on-site and offer assistance with installation, system configurations (software/hardware), troubleshooting network and printer configuration issues. This is a fee-based service and is available upon request.

Chapter 1 The Loftware Print Server

Overview of the LPS

The Loftware Print Server (LPS) is a “high end” solution for corporations that have significant product marking requirements for both bar code labels and RFID smart labels. It utilizes advanced technology that is designed to act as a middleware component to medium and large-scale business systems. It is not intended for simpler applications that can be handled with other stand-alone Loftware printing modules such as those that come with the Loftware Label Manager (LLM).

The LPS forms the basis of Loftware’s ‘Server-Centric’ approach to marking systems. Server-Centric means that all product marking in an area, building, or enterprise is controlled from a centralized server on the network. Hundreds of printers can be kept busy simultaneously, thanks to its robust, multithreaded engine.

The LPS is an extremely useful tool for printing labels from your ERP/MRP and/or WMS systems, regardless of the platform on which they reside. Host applications running on operating systems, such as UNIX, AS/400, HPUNIX, Linux, etc., can request labels through the various LPS interfaces. It is also a viable solution for client/server PC and RFID applications. LPS servers may be clustered together for fail safe redundancy as described in the ‘Clustering’ chapter.

Note: The LPS comes with the Label Design application that is found in our LLM product. This designer allows you to design your labels to meet your internal and external compliance mandates. At the time of this writing, the LPS supports over 400 thermal transfer label and RFID devices from all of the popular manufacturers.

Understanding the LPS Architecture

The “Front End”

Think of your existing Host application as the “Front End” of your labeling system. This may be a WMS/ERP application or a ‘homegrown’ application. Your database would also be considered as part of the Front End. The topography of your network may vary, but this diagram portrays the concept.

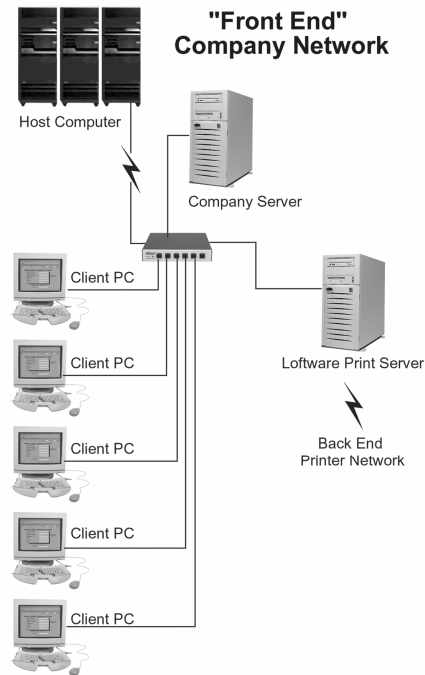


Figure 1-A: A typical Company Network showing an attachment to a Software Back-End printer network. The backend printer network is depicted in Figure 1-B.

The Front End application decides the following:

- What label format (template) to print.
- The data for the format
- The number of labels to print
- The destination printer

When these decisions are made, the label-printing event is triggered by sending a request from your application to the LPS via one of its interfaces. As explained in the next section, the interface ties the front and back ends of your system together.

The "Back End"

Think of your devices, device connections, and LPS middleware installation as the "Back End" of your marking system. Your label designs and associated files are included in this definition as well. The topography of your network may vary, but the following diagram explains the concept:

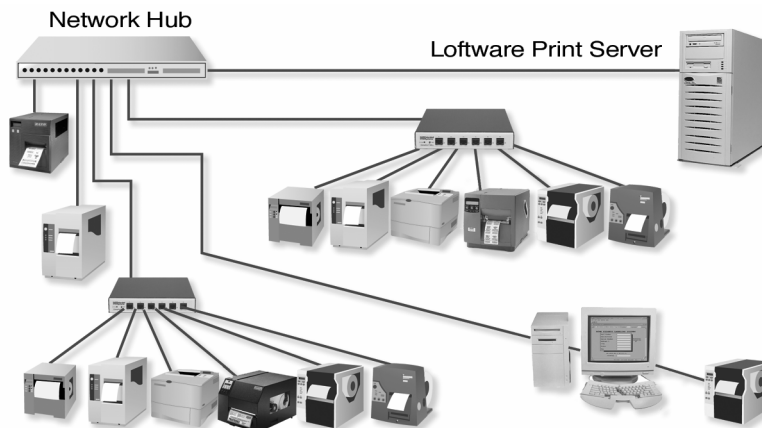


Figure 1-B: Back End of an Enterprise-Wide Bar Code Labeling System.

This diagram shows several banks of printers connected to the network, each with its own TCP/IP network address. The printers could be attached to shared workstations instead of using TCP/IP if the printer is going to be shared with other applications. Up to 500 printers can be addressed by a single LPS server.

The Interface and the Transparency Line

The Interface

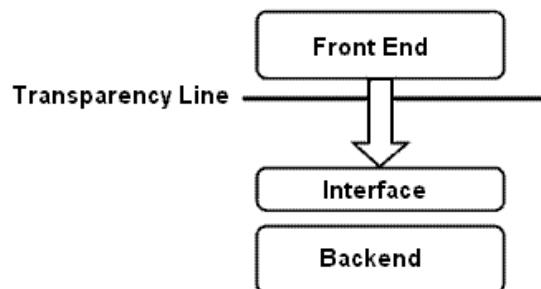


Figure 1-C: The relationship between the Front End and the Back End and the Interface

The interface is what allows your Front End Program to communicate with the LPS. The Front End makes decisions on what labels to print based on various criteria. Once this information is known, it is communicated through the interface to the Back End where it is processed. There are six different types of interfaces to choose from, each having its own unique benefits.

- The **File Interface** allows Front End applications to make requests to the LPS via a file drop to a shared network drive. The LPS detects the file and processes the request. Any program in any language can use this interface.

- The **TCP/IP Socket Interface** allows Front End applications to have bi-directional communications directly with the LPS through a socket without the need for file transfer or shared drives. Many aspects of the LPS configuration are available to the front end program through the socket, such as Label List, Field List, Printer Configs, etc. Printer status and EPC data pass back are also available in this interface. This interface is for advanced programmers only and is only available in the LPS Premier product. This interface is usually used by UNIX programmers, but can be used by C++ or C# programmers as well.
- The **Direct Socket Interface** is similar to the TCP/IP interface mentioned above. The difference is that it is much easier to code to because it is unidirectional. Print Requests are assembled in an XML file and 'shot' through the socket. No status is available other than the fact that if you cannot open the socket, the LPS probably is not running. This interface is usually used by UNIX programmers, but can be used by C++ or C# programmers as well.
- The **Connector Interface** is a Java Bean used to 'abstract' the socket communication layer from Java programmers. This makes it very easy for Java applications written on other platforms to initiate print requests to the LPS.
- The **ActiveX Interface**, the **.NET Interface**, and the **Reader Control Interface** allow programming languages to send printing requests to the LPS from anywhere on the LAN, WAN, or Internet. Under the covers, these tools actually use a socket connection for speed and reliability. These tools are available to Windows applications. See Chapters 5 and 6 for detailed information on the ActiveX Interface and the .NET interface. The Reader Control is described in the Loftware RFID Reader Solutions User's Guide.
- The **On Demand Print Thin Client** and the **Web Client** allow label requests to be made from any client PC on the network or across the Internet by prompting the operator for keyboard or database key information. No programming is required to use these client programs. See Chapters 2 and 3 for detailed information on these.

Note: For detailed information on these Interfaces, refer to the "LPS Interfaces" section in this chapter.

The Transparency Line

The 'transparency line' depicted in the figure above indicates that the Front and Back ends are transparent to one another. Applications do not need to know anything about the printers or how they are connected. Likewise, the Back End has no knowledge of what comprises the Front End or how it operates.

The transparency line is a critical factor for making the LPS effective as a "Middleware" solution. This is what has allowed Loftware to have incredible success across all applications and platforms when other systems have failed! They crossed over the line into areas that they did not understand and could not maintain or support. Our technicians, analysts, and engineers do not need to have detailed knowledge of your Front End system in order to support you.

By following the procedures outlined in the following sections, you can achieve remarkable flexibility in your system without having to "cross the line." The LPS takes care of the printing details so that our customers can concentrate on their own business processes.

Performance Considerations

The Loftware Print Server is the fastest, most advanced bar code/RFID server product on the market today. Its scaleable architecture allows it to keep many printers busy at the same time. Although the overall performance of the LPS is very impressive, printing across local and remote enterprises is a complicated process and many factors affect its performance. This section is meant to give you insight and provide you with guidelines for what you can expect using your particular system configuration.

The LPS has been extensively tested in the Loftware lab with many different scenarios, none of which may exactly match your environment. Therefore, the information in this section is of a general nature only. The actual results in your situation will vary.

Hardware Factors

Number of programs your machine is actively running

The LPS is performing many complicated tasks when it translates your printer independent requests into the native printer languages necessary to print them. The speed at which your label requests are serviced is proportional to the processor clock cycles allocated (by Windows) to the LPS. The more processor time it gets, the faster it is. In other words, if you are running other 'server' type applications like a database, fax, or email server, delays are experienced when the other servers are busy.

Number of processors

The Loftware Print Server is written to take advantage of multi-processors on a 2000/2003 machine. Because it is a multi-threaded application, the operating system balances the threads among the available processors. The net result is that the more processors you have, the more throughput performance you are able to achieve. More often than not, the bottleneck becomes the imaging and print speed of the printers themselves rather than the processing time of the LPS.

Note: The Loftware Lab has a P3, dual processor, 2 GHz machine with 525 MB of memory running 2000 Server. Thirty thermal transfer printers from various manufacturers are attached to the server as described in the "Back End" section earlier in this chapter. The performance tab on the Windows Task Manager has shown us how processing is balanced between the CPU's. Performance in the lab is quite impressive. Server Class, multi-processor machines are becoming increasingly viable as their prices continue to plummet. Serious systems should consider dedicating a server to LPS.

Available Memory

As with all programs, memory is a huge factor in speed. The more memory you have, the less applications have to swap to disk, and therefore, the faster they are. If you are running multiple applications on the same machine as the LPS, the more memory you have, the better. Do not attempt to run the LPS system on a machine with less than 64 Meg. That being said, too much memory is a waste of money. The best way to gauge memory is to use the performance monitor in the Windows Task Manager. If you see most of the memory being used, add more in 256 M blocks. Be advised that when the LPS first starts, it is at a 'baseline' memory footprint. The more printers that are printed to and the larger the print queues get, the more memory will be used.

Processor Speed

Printing throughput is proportional to processor speed. A faster processor is a better processor. Do not attempt to run the LPS on processors less than P3, 450 MHz. If the bottleneck becomes the printers imaging speed, increasing the processor speed will have no effect.

Network Speed

If you are dropping requests to a scanned network drive and/or printing to printers on your network, network speed and traffic are going to factor in. Remember, the faster the network, the greater the speed. Making printer connections across a WAN may prove to be slower than LAN connections depending upon traffic, data, and other factors. This can be avoided by dedicating an LPS server to each LAN, although your file drops may still be coming over the WAN. Real world experience has shown us that WAN speed is entirely acceptable for most applications.

Devices and Device Connections

Number of physical devices you are driving

The LPS services up to 500 different configured devices. The LPS's scaleable architecture allows you to keep all the devices working at the same time, but a performance penalty proportional to the number of devices you are driving is paid.

Note: If your system requires a large number of printers, you may divide the load between multiple LPS servers on multiple machines. For example, if you are driving 200 printers, one way you could increase performance is to have one LPS server drive 100 and the other drive the remaining 100. There is no practical limit to how many LPS servers you can have.

Thermal Transfer Printers versus Windows Printers

Using the native Loftware printer drivers for supported thermal/thermal transfer printers is always faster than using Windows printers (i.e. LaserJets). The reason for this is that we use the thermal printers' high speed, native language as opposed to sending a bitmap (picture) for Windows Printers. The LPS works with any combination of thermal/thermal transfer and Windows printers, but throughput is degraded proportional to the number of Windows printers that you make requests to.

Brand names of the printers

Loftware, Inc. maintains a position of hardware neutrality when it comes to thermal transfer printers. We do not sell or recommend printers. We do say that some printers are much faster than others are when it comes to imaging a request. Do not go by the documented print speed (inches per second/IPS) of your printers. This specification is for printing multiple copies of the exact same label without having to image new data. When purchasing printers, always look at imaging (processing) speed, not IPS.

Labeling Factors

Number of Fixed Fields on the Label

The LPS pre-downloads fixed fields (lines, boxes, fixed images, and fixed text) from your label to the printer. Subsequent requests for the same label only download the variable data that has changed. Please note that some thermal transfer printers and all Windows printers do not support this capability. The number of fixed fields on your label affects the first download of the label but

not subsequent prints. The number of variable fields and whether or not you use logos or graphics has a much larger impact.

Number of Variable Fields on the Label

Variable data fields on the label are sent to the printer every time one of them changes. If you are printing multiple copies of the same label, expect very fast throughput. If your variable data changes between labels, the system slows down because of two factors. First, the new data fields must be sent to the printer. Secondly, the printer must take the time to re-image the label with the new data before it prints.

Using Fixed Logos or Graphics

An example of a fixed graphic might be a company logo on the label that is static for all labels. Most thermal transfer printers support the pre-downloading of logos so there is a download penalty for the first download, but once the graphic is there, it is not re-downloaded. Graphics are data intensive and take longer to download than normal bar code, text, and line/box fields.

Using Variable Graphics or TrueType Fonts

If your label contains variable text fields that are formatted with TrueType fonts or graphics that change with each label, throughput suffers tremendously. Although the LPS system can handle labels like this, the time it takes to download new labels makes this prohibitively slow. Loftware strongly recommends that you use the printers' native fonts, some of which are smooth, like True Types, but are native to the printer.

Complex Formulas and Serial Numbers

If you have fields on your label that are incrementing, or have a data source defined as formula, you may pay a throughput penalty. If the printer is capable of performing the incrementing, the job is passed to it and no penalty is incurred. If, however, the printer is not capable of incrementing, then the LPS must increment the data and re-send the variable fields between labels thus causing a throughput delay.

Installing the LPS

System Requirements for the LPS

The LPS only runs on the following operating systems. The requirements set forth in the table are considered to be *minimum*. As your load increases, processor speed and memory should go up accordingly. Please see the section on performance considerations for further information on points to consider. You may also want to speak to a Loftware analyst for more information on sizing your server.

Windows OS	Processor Speed	Memory	Free Disk Space
2000/XP/2003	P3-450+	128 MB	256 MB

Note: If you are using XML with LPS, you will need MSXML version 3.0. MSXML version 3.0 is automatically installed with Internet Explorer 6.0, MDAC 2.7, and Windows XP Professional. Windows 2000 Professional and Windows 2000 Server do not have the updated MSXML version; you will have to update your version or install one of the programs listed above. If MSXML 3.0 is not an option, the legacy parser can be used.

Note also that the XML parser used within LPS only supports UTF8 and UTF16 encoding. If you wish to print an XML document with high ASCII characters (e.g., characters with accent marks), you will have to either use hex to represent the character or use UTF8 or UTF16 encoding for the character. Specifying a different encoding type in the XML will not work.

If you have questions regarding setting up an enterprise printing system, call a Loftware Systems Analyst (Professional Services Group) at (603)-766-3630. The analysis is done over the telephone and is free of charge! It is easy to start on the 'wrong foot' when designing and implementing an enterprise-wide labeling system. This can cost you valuable time and money. Analyzing your needs up front ensures that your printing system is designed properly the first time.

Before you call, please answer the questions outlined in the following section. The Analyst reviews your requirements and addresses any concerns that you have.

System Analyst Questions

Note: To help answer the following questions, please refer to the section 'Performance Considerations.' E-mailing or faxing copies of your label(s) is also very helpful.

- How many printers do you intend on driving?
- What kind of printers do you have?
- Do you use any printers with Windows Drivers like LaserJets?
- Do your labels have any fixed graphics on them such as a company logo?
- Do your labels have variable graphics on them such as a picture of a part?
- Approximately how many different label formats do you have?
- Approximately how many fixed and variable text and bar codes are on each label format?
- Do any of your labels have incrementing data fields, i.e. Serial Numbers?
- Are all of your printers in the same building?
- How often do you print multiple copies of identical labels to one printer in one request?
- How often are you requesting numerous different labels to one printer in one request?
- What PC configuration do you plan the LPS running on? (MHz, memory, processor)
- What Operating System is the server running.
- Is the PC dedicated to label printing?
- Does the PC have multiple processors?

Pre-Install System Checks

The following list of system checks helps you prevent problems that might hinder you during installation and testing. For more questions on how to implement the LPS in a specific environment, call a Loftware System Analyst to help you go over key points in your implementation strategy and/or help you architect the best solution for your needs.

Network Assessment

Assess the network speed and network protocols you plan to use to support the LPS solution. Ensure you have proper network interface cards to support required protocols and throughput. See the "Performance Considerations" section for more information on this matter. Most standard LAN/WAN configurations work fine.

Hardware Selection

Desktops/Servers/Workstations - It is important to make sure you have a PC or server powerful enough to handle your printing needs. Again, refer to the "Performance Considerations" section.

Print Servers - If you are using hub based Print Servers such as Extended Systems, HP, Intel, or Castelle to connect your printers to the network, ensure that they support the network protocols you are running and that you set their TCP/IP address. The preferred method is to use printers that already have built in Network Interface Cards (NIC's).

Printers - Select a printer that has the power and ability to support your overall needs as well as the network connections you require. We recommend printers with parallel ports or internal NIC cards.

Printer Cabling - In addition to the selection of a printer, it is extremely important to have the right cables and adapters that allow you to connect and operate the printers on your network. Identify the specific cabling requirements and read your printer manufacturer's installation guide for specifics on printer cabling for your printers.

Dedicated TCP/IP Addresses for Peripherals and PC Hardware

It is important that Servers/Desktops and Workstations running the LPS have a dedicated TCP/IP address. It is recommended that you not use DHCP as a means to configure a dynamic address. This applies for all Print Servers running in the network that support print spooling operations. All TCP/IP devices must have addresses assigned by a network administrator. Communications must be verified to all Ethernet adapters and print servers.

Operating Systems (2000/XP/2003)

2000/XP/2003 - Ensure that all services start properly and there are no hardware conflicts or service control manager errors. Also, make sure the network has properly started and your machine has an established machine account in the domain.

Print Server Setup (where applicable)

2000/XP/2003 - Using the Administrator Account, install the TCP/IP and Windows Drivers for the print servers on the Desktop/Workstation/Domain to allow you to capture printer ports during printer setup. See the manufacturer's installation guide and Chapter 2 of the LLM User's Guide for more complete instructions. This only applies to devices which do not have an internal Network Interface Card (NIC).

Use of Privileged Account

2000/XP/2003 - Installation of the Software Print Server must be performed via the Administrator Account of the local Workstation/Member Server or Domain Controller. This ensures that all registry entries on the machine are updated properly and the LPS is available to all profiles established on these machines.

Installing and Verifying

Installing the LPS

1. Read the 'Pre-Install System Checks' section.
2. Follow the instructions described in Chapter 1 of the LLM User's Guide for installation of all Loftware 'Stand Alone,' 'Server,' and 'Thin Client' programs.
3. Read and verify the install by going through the Post Install System Verification in the next section.

Note: If you have previously installed the 'Loftware Clients,' it must be uninstalled before proceeding with the LPS install.

Post Install System Verification

Here are some post-installation checks to make sure your install was successful:

1. Configure printers, labels and tags.

Note: If you want to use the Label Design Module from a PC other than where the LPS is installed, it is necessary to install it on a separate PC. Loftware recommends that additional copies of Loftware Label Manager be purchased for design purposes.

Configure your printers from label design mode on the server. It is helpful to keep the following points in mind as you do this.

- Hardware DIP switch settings - See Owner's Manual.
- Settings for direct thermal/thermal transfer, stock type, and baud rate where applicable
- Print head temperature settings
- Label stock calibration tests
- Label home position set and tested
- PC port (if connected to a port) communication tests to printer - See Manufacturer Installation Guide for examples. Loftware recommends that all printers have a TCP/IP address.

2. LPS Service configuration

By default, the LPS Service is installed with the Administrator account as the primary Manager of the service. This account can be changed if desired. Changing this account is done via the Service Control Manager that is accessible through the Control Panel. The service is configured to start manually by default, but can be set to auto-start at system boot.

3. Check LPS account privileges

If you are running from the Administrator Account, no action is required. If, however, you are running from a custom defined account (not administrator):

- Account must have administrative rights to the local machine.
- Account must have administrative rights to the directory receiving requests.
- Account must have the right to logon as an NT service.

4. Check your ability to run batch print (LLMWBP32.EXE)

Try to invoke the program “C:\Program Files\Loftware Labeling\llmwbp32.exe” from the desktop. If you can successfully run the program without errors then your account is operating properly and no further action is required.

From a custom defined account (not administrator):

- a. Log out as the administrator.
- b. Log into the system using the username and password chosen during the LPS setup.
- c. Try to invoke the program “C:\Program Files\Loftware Labeling\llmwbp32.exe” from the desktop. If you can successfully run the program without errors then your account is operating properly.
- d. Log out and log back in as administrator.

What to do if you have errors invoking Batch Print:

- a. Make sure the account you are using for LPS is a member of the administrator group.
- b. Check the DCOM privileges for llmwbp32.exe.
- c. Once these issues have been checked, logout and log back in to the account and try to run LLMWBP32.EXE from the desktop.

5. LPS Configuration Utility

Make sure that your scan directories, file name extensions, and LPS settings are correct for your implementation. See the section on the LPS Configuration Applet in this chapter for more detailed information.

6. Network file shares established

Ensure you have established all proper file share permissions for all external users who are writing and moving files that are picked up by the file interface of the LPS. A hidden share called ‘LOFTWARE\$’ is created during the installation process. See your system administrator for more information on file share permissions. The LOFTWARE\$ share may not be created during installation if your security settings prevent it. If this is the case, you can manually create the share by right clicking the ‘Loftware Labeling’ directory in Explorer and choosing ‘Sharing.’

7. Create test labels/tags and test files for the LPS

- a. Configure your printer using Label Design Mode.
- b. Create a simple label/tag with two variable fields.
- c. Ensure label specific options are set correctly for your printer.
- d. Test print a label from design mode to be sure that the printer and the LPS are configured properly.

Once the label is configured properly:

- e. Create a small .PAS, .Xml, or .CSV file as described in this chapter.
- f. Be sure to start the LPS service as described later.
- g. Specify the printer name or number in the file.
- h. Ensure your label format is on the system that is running the LPS.
- i. Send your pass file to the scan directory you specified. The LPS should pick it up and print the request.

Configuration of the LPS

Loftware has provided a handy configuration applet that makes it easy for you to start, stop, and configure the LPS. Use one of the following techniques to launch the applet.

From the Windows 'Start' button

Start > Programs > Loftware Labeling > Print Server > LPS Configuration

From the Windows Control Panel

1. Start > Settings > Control Panel
2. Double click the "Loftware Print Server" icon from the Control Panel screen.

Note: The 'Status' and 'Notifications' tabs are associated with the 'Thin Client Status Program' and the 'Loftware Notification Agent.' They may be disabled (gray) if you do not have a license for them. These programs and how to configure them are documented in the following chapter.

General Tab

The purpose of the General tab is to set up scan directories for use with the File and ActiveX Interfaces, described in the next section of this chapter. As you can see in the following dialog, the LPS defaults to scanning a directory called 'WDDrop,' which is a directory under the primary Loftware directory. This should be satisfactory for most labeling systems. You may, however, specify a different network directory or set up multiple scan directories if you feel that your system warrants it.

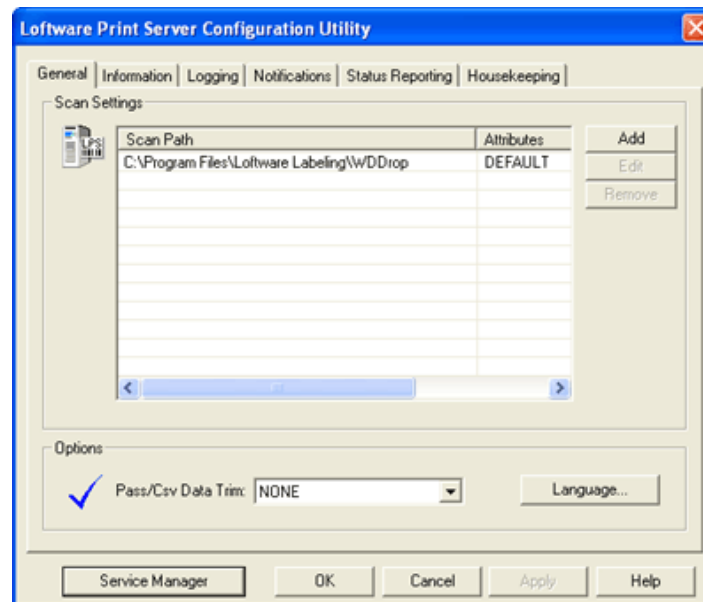


Figure 1-D: The LPS Configuration Utility showing the General tab

Scan Settings Section

Multi-Scan Grid – This is for the ‘File Interface’ only. Consider the following scenario: There are 50 printers controlled with the Loftware Print Server. Requests for labels happen in both ‘batch’ and ‘on demand.’ Let us say that your UNIX server has made 500 different label requests for 40 of the printers. At the same time, an ‘on-demand’ operator wants to print one label on printer 41. If you were scanning only one directory, there may be a considerable delay before the single request is processed, because of the hundreds of requests that are in queue before it.

There are several ways of setting up scan directories for optimizing system performance. One way is to have a separate scan directory for every printer, and drop file requests to the appropriate directory depending on the target printer for which the request is made. All scan directories are serviced simultaneously, so that in the above scenario, each single request buried under multiple large requests is serviced quickly and with no noticeable delays. This is referred to as multi-threaded input and although easy to set up, it represents an advanced feature found only in the Loftware Print Server. Another method that would achieve similar results in the scenario given above is to set up just one more scan directory. Use one directory for your batch requests and the other for on-demand requests.

The concept of ‘multi-scan’ is not an exact science. The best advice we can give is to carefully consider all aspects of your labeling system and create scan directories based on a logical division of tasks.

Note: Simple systems that do not have high throughput requirements can simply use the default of one scan directory.

Adding a Scan Directory

Before adding a scan directory, you must first create it using Windows Explorer.

- Click the ‘Add’ button in the LPS Configuration Utility (General Tab) The following dialog box is displayed:

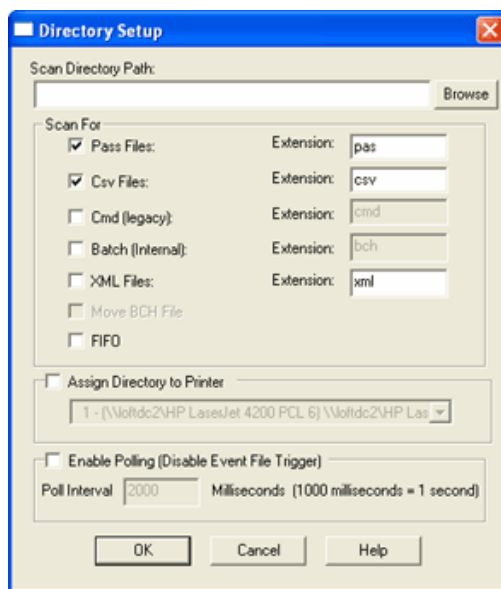


Figure 1-E: Directory Setup Dialog Box

Scan Directory Path - This is the directory into which print requests are dropped using the ‘File’ Interface. The directory can reside on any network drive to which both your application and the LPS have Read/Write access. If you are using the LPS in ‘Service’ mode without logging on, UNC paths like ‘\\remote\server\dir1’ are required, but only if the scan directory is on another PC. If the scan directory is local, a UNC path is not required.

Use the Browse button to locate the scan directory. Using the Browse button eliminates possible errors when typing the scan directory path manually. The directory must exist before you can actually browse to it. It *is not* created for you. You may create scan directories using file Explorer.

Scan For Section

Scan for Pass Files – The LPS scans for .pas (default extension) files. Fill the extension with the extension for which you wish to scan. (i.e.*.pas). Pass files are documented in the file interface section of this chapter.

Scan for CSV Files – The LPS scans for .CSV (default extension) files. Although harder to read, CSV files are much more compact than .pas or .xml files. CSV files are documented in the file interface section of this chapter.

Scan for Xml Files – The LPS scans for .xml (default extension) files. This scans for files that have a header row and subsequent data lines. Xml files are documented in the file interface section of this chapter.

FIFO – Acronym for “First In, First Out,” which means that when checked, files are processed in the order in which they are received, and not the order in which they are entirely written to the sectors on the hard disk. FIFO is not guaranteed unless this setting is turned on. Expect to pay a performance penalty due to the sorting that must occur. Usually any penalty of this type is negligible and can be ignored.

Scan for Command Files and Batch Files – Legacy settings, don't use!

Assign Directory to Printer Section

This assigns a designated printer to the added directory. This is helpful if it is difficult to include a printer number command in your request. Any files dropped into a directory with this setting turned on are dispatched to the assigned printer regardless of any reference in the file itself. Extra care must be taken to ensure that the file you are requesting is indeed designed for the assigned device.

Enable Polling Section

Polling - Polling must always be enabled for shared network drives. The LPS remains idle until the operating system notifies it that the contents of the scanned directory have changed. If you have polling disabled and find that the LPS is not processing your files, enable polling and set the polling interval to the milli-second interval that you wish to poll the directory; 1000 mSec (1 second) is the default.

Options Section

Pass/CSV Data Trim - Valid for Pass Mode only. If your data contains leading or trailing spaces, you can use this setting to trim them. Choices in the drop-down list are:

None – No leading or trailing spaces are trimmed. This is the default.

Trailing – Trims trailing spaces only

Leading – Trims leading spaces only

Both – Trims both trailing and leading spaces

When would you set it to “Both”? *Example:* You have a label that is connected to a database, and the key field on a label has either a leading or trailing space(s) in it. In this case, the database data fields connected to that key field are not found because the program is searching for the key field without the spaces. In this case, selecting “Both” ensures that no spaces are included in the field.

Language Button

As stated in the LLM User's Guide, selection of the default language for Loftware Applications (Design, On-Demand, Range, Batch, etc.) can be made on a "per-user" basis. However, when a service such as the LPS is being used, the language change is for the service, not for the user; therefore it becomes the default language for all users.

Example: A user logged in to a machine whose default language has been set to "English" would like the language of the LPS Service to be displayed in their native language of French.

To set the default language for the service to French, the following steps are performed:

1. Open the Loftware service
Start > Programs > Loftware Labeling > Print Server > LPS Configuration
2. Press Language.. under the General Tab.
3. Select French (Français) from the drop-down list.

Remember, this changes the default language for ALL Loftware users on this machine to French, and also changes the language for ALL the other Loftware services (Web Client, Notification

Agent) on this machine as well. This setting controls the user interface only and does not affect LPS performance.

Logging Tab

The logging features of the LPS can be quite useful for debugging purposes should something go wrong. It is important to note that, when run as a service, the LPS does not have access to on-screen dialog boxes for displaying error messages. Once your system is polished and 'on-line,' this is of little consequence. However, during the debugging stage of your system, it is imperative that error and warning messages can be read. Logging assists you in doing this. There are three types of logging: Trace, Event Logging, and On Screen Logging for Interactive Mode.

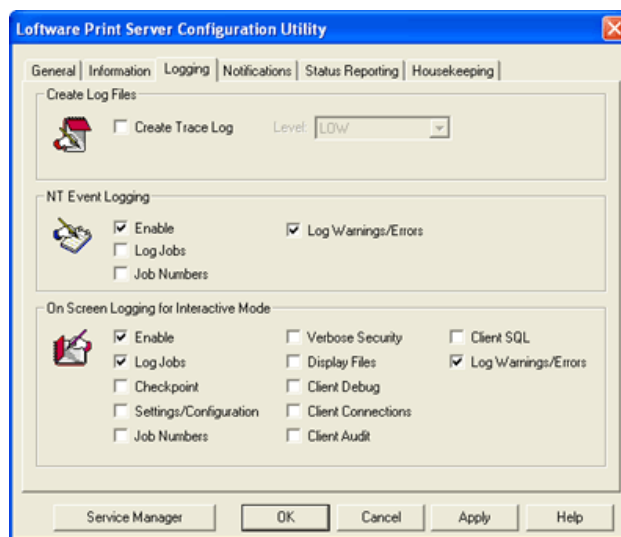


Figure 1-F: Event Logging Tab

Create Log Files Section

Create Trace Log - Do not turn this option on unless directed to do so by Software Support Services. If the LPS is running in interactive mode, this option creates trace.txt and trace1.txt files in the temp directory defined by Windows. If the LPS is running in service mode, these files are created in the Windows directory. All commands and internal processes handled by the LPS during your session are logged to these files. The amount of information logged is determined by the 'Level' setting. 'Low' logs the most information; 'High' logs the least; if you cannot find your trace files, use the Windows 'Search' utility to locate them.

These files can be very helpful when something has gone wrong and you have contacted Software Support Services for assistance. Once your system is up and running, it is imperative that this option be turned off; otherwise, you are eventually going to run out of disk space.

Please note that the information contained in the trace log is for the exclusive use of Software Support Services and is of no value to you. Use NT Event logging and/or On Screen Interactive Logging for your own monitoring and troubleshooting purposes.

Event Logging Section

Unlike the trace log, this type of logging is more useful for your own monitoring and debugging efforts. All warning and error messages are written to the NT event log. This is very helpful when running the LPS as a service because a service cannot display messages on the desktop. The log can be accessed through the event log viewer under '**Start > Programs > Administrative Tools > Event Viewer.**' Items are logged to the event log whether you are running in service or interactive mode.

Enable - Enables the logging of informational events to the event log. The exact type of information written to the log is determined by the log command settings described below. Remember, error and warning messages are always written to the log regardless of whether or not this setting is on. This setting also allows access to information about who last changed the configuration of the LPS and when this change was made.

Log Jobs - The log Commands, Batch, CSV, and PAS settings write an entry to the event log each time one of these files is successfully processed. Once your labeling system is operational, it is recommended that these settings be turned off. A quick way to do this is to simply uncheck the 'Enable' setting. The other settings are remembered for the next time you decide to enable logging.

Note: Make sure that the maximum size limit of the Event Viewer is set appropriately so that the log does not continue to grow until all disk space is consumed. This is done in the event viewer itself.

Job Numbers - The printing sub-system returns a unique job number back to the LPS for every submitted job. Turn this option on if you wish these numbers to be recorded in the log.

On Screen Logging for Interactive Mode Section

Interactive Logging is only applicable when the LPS has been launched in 'Interactive' mode from the Start Menu. The selections made here are displayed 'real time' in the status window of the LPS. If this setting is disabled, a minimal number of 'top level' messages are still displayed. It is recommended that On Screen Logging be enabled with the appropriate settings described below. Only enable the settings that you are interested in; otherwise, you are 'flooded' with log entries about which you may not care.

If enabled, positive acknowledgement of the processing of file interface (such as CSV, PAS, and Xml) requests is seen. Enable the one that you are using. If you want to view requests coming in from the ActiveX Client Control, enable the 'Log Pass' checkbox.

The middle row of checkboxes displays different types of information that may be helpful in troubleshooting. If you are having problems with the LPS, try enabling these settings. The logged information may provide the clue that you are seeking.

The client checkboxes are useful in monitoring thin client activity. If you are not using any of the Loftware Thin Client applications (Chapter 2), then these settings are of no use. If you are using thin clients, the logging of this information can be quite valuable in understanding/troubleshooting client activity.

Note: The WDLOG utility uses the Interactive Mode log settings.

Notification and Status Reporting Tabs

Note: The 'Status' and 'Notifications' tabs are associated with the 'Thin Client Status Program' and the 'Loftware Notification Agent.' These tabs are disabled if you do not have the Premier Edition of the LPS. These programs and how to configure them are documented in 'Thin Clients' chapter.

Housekeeping Tab

Housekeeping has taken the place of the "Directories" tab in Loftware Versions 5.5 and above. The purpose of Housekeeping is to allow you to set a time when the levels of disk space used by Printed Jobs and Job errors (in MB) are checked. Printed Jobs and Job Errors are saved into various working folders, and while it is helpful to keep these 'old' job files around for a period of time for auditing purposes, they must be deleted or 'purged' eventually in order to minimize disk space and keep label printing at a high speed. Sections of this tab are described below.

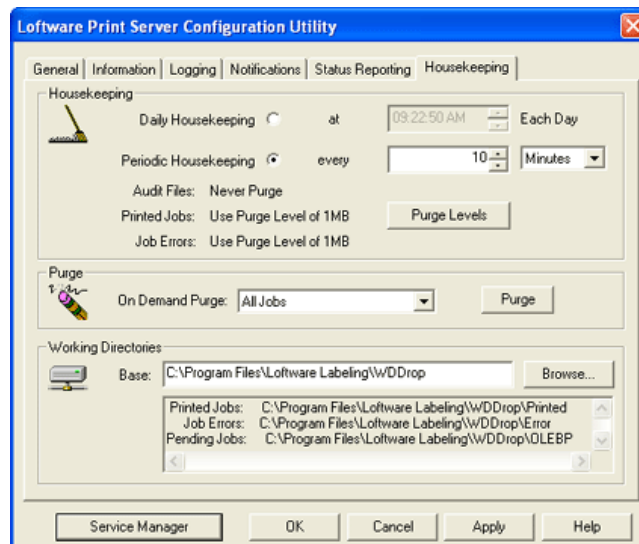


Figure 1-G: Housekeeping tab

Housekeeping Section

Daily Housekeeping – If Daily Housekeeping is selected; then the Periodic Housekeeping option is disabled. This allows you to set the time of day *each day* that you would like your Purge Levels to be checked. The editable fields are Hour, Minute, Second, and AM/PM (HH:MM:SS AM/PM). If you run the LPS for 2 consecutive work shifts for example, you may want to set the housekeeping time for 1:00 a.m., when production is not taking place. The default time is 1:00 a.m. daily.

Periodic Housekeeping – When this option is selected, housekeeping is not performed daily, but is checked when the interval time set in minutes, hours, or days has been reached. The interval time is set to check the Purge Levels of "Printed Jobs," "Audit Files," and "Job Errors" to see if they need to be reset and purged. The default time is 10 minutes. If the Purge Level has been reached,

the selected file categories are checked and purged if necessary. This is only a **check** to see if the values set in the drop-down lists for “Printed Jobs” and “Job Errors” have been met or exceeded. If they have not, then no purging takes place. The Periodic Housekeeping time can only be set when the Daily Housekeeping option is NOT checked.

Purge Levels - Saving old jobs is helpful for auditing purposes; however, in the interest of not running out of drive space, they must be deleted at some point. This is where the purge settings for both the Printed Jobs and Job Errors directories are useful. If the *Use Purge Level* option is chosen, when the space consumed by the files in a directory reaches the designated purge level (a number set in Megabytes), disk space is recovered by deleting the oldest files first until the size of the directory has been reduced to 75% of the purge level. If the setting is 1 (default), then 250k of the Printed Jobs and/or Job Errors files are deleted when they reach the 1 MB level. Remember that these purge levels are evaluated **ONLY** when the Evaluation Interval time is reached. Because of this, the space actually consumed in the directories may be slightly higher than the purge setting before the purge takes place.

If you set the Purge Level to *Never Purge*, it is your responsibility to delete the files yourself at the appropriate time. If you neglect to do this, you are going to eventually run out of disk space, which has been known to have catastrophic effects.

Setting to *Always Purge* deletes all of the files in the directory immediately after processing. Any extra disk space consumed by buffering old files is eliminated; however, you no longer have the ability to view history or job errors and resubmit jobs using the Status program.

Purge Section

On Demand Purge – The drop-down list allows you to check the directories to purge. Select the directory you would like to purge and click on the Purge button. This option is useful if you wish to perform a manual purge.

Working Directories Section

By default, the LPS creates three ‘working directories’ under the ‘**Program Files\Software Labeling\WDDrop**’ directory. They are ‘Printed,’ ‘Errors’ and ‘OLEBP.’ These are the only directories used by the LPS regardless of how many directories you are scanning or the number of printers that you are driving. The Browse button is provided if you need to change their locations.

Note: The only reason to change the ‘Working Directories’ location is if you want to centralize them for another LPS server to use in an auto-failover scenario. For more information on this, refer to the chapter on Clustering and Auto-Failover later in this User’s Guide.

The following points illustrate some of the ways that these directories are used by the system:

- All requests coming through any interface are renamed and saved off into the appropriate directory depending on the outcome of the job.
- Jobs originating from the TCP/IP interface are saved in the format specified in the protocol header when submitted. The extension of these files is changed to **.int**.
- Although the syntax of file interface requests created in .CSV, PAS, and Xml are not changed, their names are. The most efficient way to view the files is with the Status program, which is documented in the following chapter.

- Print requests that have printed without error are saved in the Printed directory along with a corresponding .lpj file. The .lpj files contain information about the job and are for internal system use only. Under normal operating conditions, the Printed directory grows the fastest and is purged according to the housekeeping settings.
- Requests that encounter critical error conditions are saved into the Errors directory. An example of a critical error would be a syntax error in your pass file, referencing a label that does not exist, etc. Other errors like 'Printer out of Labels' is placed in the OLEBP directory and scheduled to retry until the job prints. No user interaction is required for these types of errors.
- Jobs that are ready to print but have not actually been processed by the print engine are 'buffered' in the OLEBP directory until they are sent. This allows the LPS to auto recover any jobs that were not printed after a system crash or unscheduled shutdown.
- The only way to prevent the LPS from auto-recovering is to shut it down and use File > Explorer to manually delete the jobs in OLEBP.
- Jobs in the Printed and Errors directory can be resubmitted by right clicking them in the Status program (Chapter 2) and selecting 're-submit.' Any job that is resubmitted is copied to a new name in the printed directory and has an **.rsd** extension.

LPS Modes

Now that you have successfully installed and configured the LPS, it is time to start it up and make a test request through one of the interfaces. There are two ways to start the LPS: as a **service** or **interactively**.

Service Mode

The LPS is designed from the ground up to run as an NT Service with all applicable security. An NT service is an application that can be configured to run when the PC is booted. It runs at a lower level (ring) than ordinary programs and cannot be seen or changed by the operator.

Services can perform their functions without requiring the operator to log onto the server, thus providing protection from intentional or accidental change (security). It is recommended that you run the LPS as a service when in production.

Two methods can be utilized to start the service. Each method allows you to set the service to 'Auto Start' at system boot so that you do not have to remember to start it each time your system is rebooted. The service defaults to manual start. Be sure to invoke the configuration applet before starting the service to configure it properly. Don't switch to auto start until you have finished verifying your system. The reason for this is you may want to flip back and forth between Service and Interactive modes while testing and debugging your system.

***Note:** Since services cannot display error or status information on your desktop, starting and testing your labeling system in 'Interactive' mode is recommended until you are confident that all aspects of your system are performing to specification. At this point, make sure you are logging critical error information to the Event Log. This is explained in the 'Configuration of the LPS' section.*

Start the LPS in Service Mode

To start the Service, do *one* of the following:

- Advanced users may navigate to the Service Control Manager and start and stop it from there, or..
- Press the 'Service Manager' button in the LPS configuration located in the Control Panel. Select 'Loftware Print Server' as the service to start, and press the Start button.

Interactive Mode

The **Start Menu** is used to run the LPS interactively. When the LPS is run in interactive mode, its status window allows you to monitor what it is doing at any given moment. The Options Menu has a "Security Checks" choice that performs a diagnostic scan of the LPS installation. This is very helpful for troubleshooting any problems you may encounter while setting up your system. Use the 'Logging' tab of the control panel applet to turn on extra debugging information that can be seen on the screen while running in this mode. Turn the logging off before bringing your system to a 'live' state.

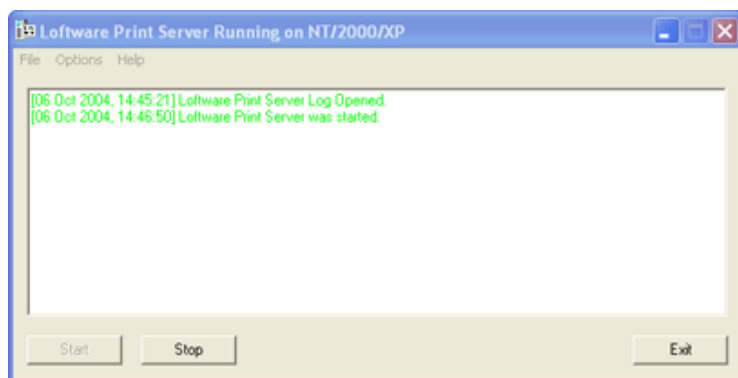


Figure 1-H: LPS running interactively on NT

Note: If you are running the LPS as a service, you must stop the service before invoking it interactively. There is no reason to do this unless you are having problems and wish to do a diagnostic scan or take advantage of On Screen Logging.

- The LPS can be launched using the 'Start' command: Start > Programs > Loftware Labeling > Print Server >LPS interactive.
- The LPS configuration utility can be launched from the Options | Configuration menu. Check the configuration settings before clicking the "Start" button for the first time. The configuration utility is documented in the previous section.
- The Options Menu has a "Security Checks" choice that performs a diagnostic scan of the LPS installations. This is very helpful for troubleshooting any problems you may encounter while setting up your system.

You must press the "Start" button to tell the LPS to begin accepting data from its interfaces. If you launch it with a "-run" parameter, it automatically comes up scanning.

Print Request Data Structures

There are 3 data structure syntaxes which can be used to make requests to the LPS. Any of these data structures can be used regardless of the interface being used unless otherwise noted in the interface section.

Once the front end application has decided:

1. Which label template to print
2. What data to place in the template
3. Label Quantity
4. Target Printer

It will place this data into a PASS, CSV, or XML structure and pass it to the LPS through the interface. Many of the examples in this section assume that the File Interface is being used. The other interfaces still apply, you would just send the data structure through the interface (like a socket) instead of creating a file.

Pass and CSV Commands

All LPS Pass commands start with an asterisk (*). Each Pass file must start with the *FORMAT command, and end with the *PRINTLABEL command. The following commands apply to the .PAS and .CSV file syntaxes. The commands for Xml are the same with the exception of starting with an ‘_’.

Valid LPS Commands

*FORMAT	Specifies which label to send data to. This is the name of the label designed in label design mode. If no path is given, the default ‘labels’ path of the LPS is assumed. If no extension is given ‘.lwl’ is assumed. If your labels are shared on a network drive, include a full path.
*JOBNAME	This names the job for status reporting purposes. If you do not name your job, it is displayed in the status view as the name of your pass file or batch file with an incrementing number. Using this setting allows you to find your job in the Status Client or to check the status of a job through the Socket Interface.
*PRINTERNAME	This allows you to use a printer alias instead of a number. For example, instead of using “*PRINTERNUMBER,4”, you could use “*PRINTERNAME, Shop Floor Printer.” This is much more descriptive, but it does require that you give each printer in your system an alias. (See Chapter 2 of the LLM User’s Guide on how to do this.) Also, if you are using *PRINTERNAME, you do not use *PRINTERNUMBER. The latter always takes precedence over the former.
*DELINKODBC	Overrides the database. You must provide all data for the labels. This is helpful for when you normally ‘push’ the data to the LPS for your production, but the fields are connected to a database for backup purposes using the On Demand Print thin client. Include this command in your .pas file to turn off the database links for normal production.

- *SENDFORMAT,0** Suggests that the LPS suppress the sending of the format information to the printer. The printer recalls the format from memory and throughput time is increased. This only works on printers that support the pre-downloading of the fixed fields on your format.
- *PRINTERNUMBER** Specifies to which configured printer to print (1-499). If you do not supply an alias for your printers, you must use their number.
- *QUANTITY** Specifies the quantity of labels to print.
- *DUPLICATES** Specifies how many copies of the label to print. This is useful if you are using incrementing numbers and you need to print multiple copies of the same number before incrementing it. For example, you may need to put two labels on a box with the same serial number - one label for each corner.
- *PAGES** Specifies how many pages to print. This is only used if you are using a page layout for your label. See Chapter 9 of the LLM User's Guide for information on layouts.
- *PRINTLABEL** Prints the current label. Any commands after this are considered a new label. Your label will not print without this command
- *TRAY** In Software Versions 5.3.2.12 and above, a *TRAY command is available for Laser Printers only. If the printer you are printing to has more than one tray, this command is designed to send a label to a particular tray on the printer. To select a tray, send a “*TRAY, tray” command in the pass file where **tray** is the NAME of the tray that supplies the paper. The name **MUST** match the configured tray name in the driver setup (not case sensitive).

The *TRAY command is NOT sticky by default; in other words, if you send a stacked pass with a *TRAY command in the first request, subsequent requests do NOT print to the same tray (unless the default tray was selected), as it is 'OFF' or set to '0' by default. To make the *TRAY command sticky, set the following in the llmwdn32.ini file:

[Options]

StickyTray=1

lets a specific tray to which the label is printed. Tray data must match the selections in the printer driver's properties box.

Pas and CSV Syntax

The Pass syntax is a simple ASCII structure that has all the commands, field names, and data for the labels that are to be printed. The field names in the file match the ones given to their corresponding fields on the label during design time. Each Pass file can have many different label requests within it. Once constructed, this file is passed to the LPS through the interface, which is usually a file or a socket.



A “Lights the Way” Tip

Providing Data for the General Case

You can pass fields in your **pass (.pas) file** that the label does not use.

This is useful because you can provide your user with a list of legal field names that you include in the pass file that users may design new labels for. These labels work with your system, preventing the need for changing the source code.

The label called out in the ***FORMAT** line pulls only the fields it needs from the pass file, ignoring the rest.

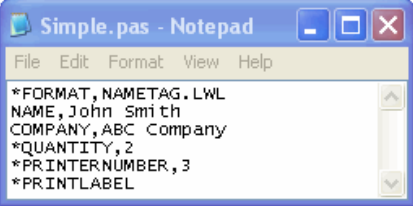
See the ‘Providing Data For the General Case’ section for information on how to do this.

Note: If you are using the alternate syntax described in the examples below, you may use the following commands as alternates to the ones above. The reason for this is to make it easier for report generation software to create the file. Note that the “*PrintLabel” command has no alternate because it is not used in the CSV syntax. **FORMAT, JOBNAME, PRTNAME, PRTNUM, LLMQTY, LLMDUP, PAGE.**

Note: If the above alternate commands do not work for you, you may specify others by creating a section in the llmwdn32.ini file called [**CommandRemap**] in your Windows directory. In this section you may specify - [**CommandRemap**] **FORMAT = yourFormatCommand**
***PRINTERNAME = yourPrinterNameCommand** etc. This command only works with .csv files and a header. It is not designed to work with .pas files.

Example 1: Simple Pass File

This file loads a format called “nametag.lwl” that was previously designed in Label Design Mode. The next two lines specify data for variable fields that you named “name” and “company” when the label was designed. The next lines instruct the LPS to print two of these labels on configured printer #3.



```
Simple.pas - Notepad
File Edit Format View Help
*FORMAT, NAMETAG.LWL
NAME, John Smith
COMPANY, ABC Company
*QUANTITY, 2
*PRINTERNUMBER, 3
*PRINTLABEL
```

Figure 1-1: Simple Pass File

Alternative syntax: This file could also be represented as a CSV file with a header as:

```
"FORMAT", "PRTNUM", "LLMQTY", "NAME", "COMPANY"
"NAMETAG.LWL", "3", "2", "JOHN SMITH", "ABC COMPANY"
```

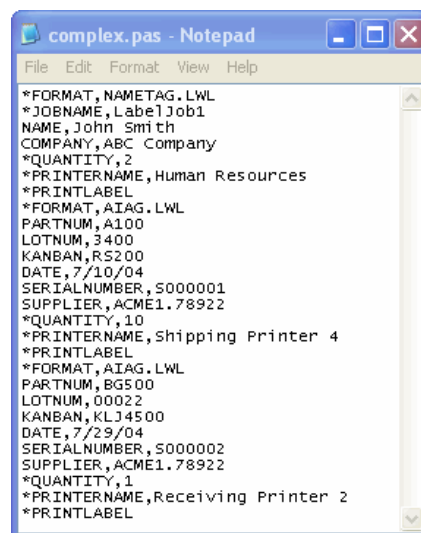
Note: For .csv files, each new record (label) is added on a new line of the above file. The header on the top line is only needed once. The amended example below shows multiple record requests in one file. Each label uses the same label format but prints to a different configured printer and has different data. Notice that the alternative command names are used in the header record (line 1).

```
"FORMAT", "PRTNUM", "LLMQTY", "NAME", "COMPANY"  
"NAMETAG.LWL", "3", "2", "JOHN SMITH", "ABC COMPANY"  
"NAMETAG.LWL", "2", "1", "DANA SCULLY", "X-FILES COMPANY"  
"NAMETAG.LWL", "3", "20", "FOX MULDER", "FBI AGENCY"
```

Example 2: More Complex Pass Files

This example illustrates that you can “stack” requests in a pass file. It also shows how advanced users might use the *JOBNAME and *PRINTERNAME commands. If you need to stack more than 50 requests, your file is much smaller using the alternate syntax following this example.

Also, please note that this example uses different label formats and therefore different field names for some of the requests.



```
complex.pas - Notepad  
File Edit Format View Help  
*FORMAT, NAMETAG.LWL  
*JOBNAME, LabelJob1  
NAME, John Smith  
COMPANY, ABC Company  
*QUANTITY, 2  
*PRINTERNAME, Human Resources  
*PRINTLABEL  
*FORMAT, AIAG.LWL  
PARTNUM, A100  
LOTNUM, 3400  
KANBAN, RS200  
DATE, 7/10/04  
SERIALNUMBER, S000001  
SUPPLIER, ACME1.78922  
*QUANTITY, 10  
*PRINTERNAME, Shipping Printer 4  
*PRINTLABEL  
*FORMAT, AIAG.LWL  
PARTNUM, BG500  
LOTNUM, 00022  
KANBAN, KLJ4500  
DATE, 7/29/04  
SERIALNUMBER, S000002  
SUPPLIER, ACME1.78922  
*QUANTITY, 1  
*PRINTERNAME, Receiving Printer 2  
*PRINTLABEL
```

Figure 1-J: Complex Pass File

Example 3: .CSV File

Alternative Syntax: This File could also be represented in ASCII CSV file format with header as:

```
"FORMAT", "PRTNAME", "JOBNAME", "LLMQTY", "NAME", "COMPANY", "PARTNUM", +  
"LOTNUM", "KANBAN", "DATE", "SERIAL NUMBER", "SUPPLIER"  
"NAMETAG.LWL", "Human Resources", "LabelJob1", "2", "John Smith", +  
"ABC + Company", "", "", "", "", "", ""  
"AIAG.LWL"Shipping Printer 4", "", "10", "", "", "A100", "34000", "RS200", +  
"1/10/00", "S000001", "ACME1.78922"  
"AIAG.LWL", "Receiving Printer 2", "", "1", "", "", "BG500", "00022", "KLJ4500", +  
"3/12/98", "S000002", "ACME1.78922"
```

*Note: The + at the end of the line means this line is continued on the next line. Your file does not contain the plus or wrap to the next line. You only go to a new line when you are starting a new label. Also, note that data fields not used by a particular label are empty (“”). Quotes are not necessary for the data, but if your data contains commas, they are mandatory. Also notice that the above example uses the alternate commands for *PRINTERNAME etc. and that the alternate syntax is much more compact.*

Xml Files

Note: The XML parser used within LPS only supports UTF8 and UTF16 encoding. If you wish to print an XML document with high ASCII characters (e.g., characters with accent marks), you will have to either use hex to represent the character or use UTF8 or UTF16 encoding for the character. Specifying a different encoding type in the XML will not work.

Xml Commands

All commands begin with an underscore '_'. Commands are 'Sticky' which means that the Xml header defaults are always used unless a request header overwrites any of them later in the .xml file. Creating a 'Stacked' (multiple requests per file) .xml file for one particular label format only requires one _FORMAT command entry in the Xml header.

<code>_FORMAT="Label1.lwl"</code>	The _FORMAT command references the label format to use. If no path is specified, the 'LabelsPath' entry in Options File Locations is used. If the file is missing an extension, the default '.lwl' is assumed.
<code>_JOBNAME="SampleJob001"</code>	Optional entry used to identify the current job during the printing process. If _JOBNAME is not present, the filename is used.
<code>_QUANTITY="1"</code>	Optional entry used to designate the quantity of labels for print. If _QUANTITY is NOT present, a default of 1 (one) is assigned.
<code>_DUPLICATES="1"</code>	Optional entry used to designate the number of pages to print when a label has a layout. There is no effect if no layout is used.
<code>_PRINTERNUMBER="1"</code>	Required entry used to designate the printer to which the label is printed. The _PRINTERNUMBER value corresponds to the Printer number in Configure Printers dialog. OR...
<code>_PRINTERNAME="PrinterAlias"</code>	Where PrinterAlias is the Alias name assigned to the printer in Configure Printers Connection dialog. Only one entry, PRINTERNUMBER or PRINTERNAME is needed.
<code>_TRAY="1"</code>	Optional entry used to designate the tray to print to when printing to a Laser Printer. Tray data must match the choices in the printer driver's properties box.

Xml File Syntax

Xml files that will sent to the Software Print Server must begin with the following two lines.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "C:\Program Files\Software Labeling\Batch\label.dtd">
```

This is to ensure that your .xml file is using the Loftware default label.dtd file.

Xml Header

The Xml Header is a beginning tag in which you define the default commands for the entire .xml file.

- The Header starts with **<labels** and is followed by any default commands you wish to set separated with a space.
- End the header with a '>' If no commands are set within the header, then they must be set in the beginning tag of each individual request.

Example Xml Header

```
<labels _FORMAT="Label1.lwl" _JOBNAME="SampleJob001" _QUANTITY="1"
_PRINTERNUMBER="1">
```

OR- without any commands

```
<labels>
```

Xml Requests and Commands

- Requests can be in any order, first entry per request is a request header **<label>**
- You may overwrite any of the default commands set in the Xml header or add more commands to this one request.
- If no commands are set here, defaults are used.
- The final entry per request is a **</label>** line
- Multiple requests are possible and are separated by the end of one request **</label>** and the beginning of the next request **<label>**.

Note: Remember, when adding commands separate each with a space ' ', also note that any default commands overwritten here are for this one request only.

Example Xml Request Header

```
<label>
```

The above would use the defaults listed in the Xml header **-OR-**

```
<label _QUANTITY="5" _PRINTERNUMBER="2">
```

This one request would use **_FORMAT** from the Xml header but its own **_QUANTITY** and **_PRINTERNUMBER**.

Xml Data

- All data lines begin with '**<variable name='** followed by the field name in double quotes followed by '**>'** then the data for the field and finally an end tag '**</variable>'**
- If a field does NOT appear in the Xml file, the data is cleared and NOT remembered across labels.

Example Xml Data Line

```
<variable name="Text0000">New Data</variable>
</label>
```

Marks the end of the current request definition.

```
</labels>
```

Marks the end of all requests in the entire xml file.

NOTE: If you are unsure if you have written your xml file correctly, open the xml file using Internet Explorer 5+. If the file opens without error, your xml file is correct and is processed when dropped to the scan directory. Please also note, to open a Loftware based xml file you must have the Loftware standard label.dtd file in the same directory as your xml file, or, as seen at the top of this file, in the DOCTYPE tag, you can define where the label.dtd file is located (this defaults to \Loftware Labeling\Batch).

Example 1: Simple Xml File

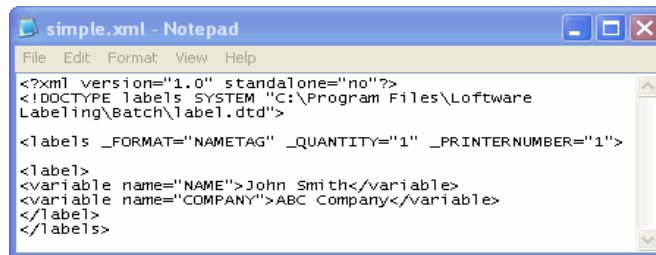
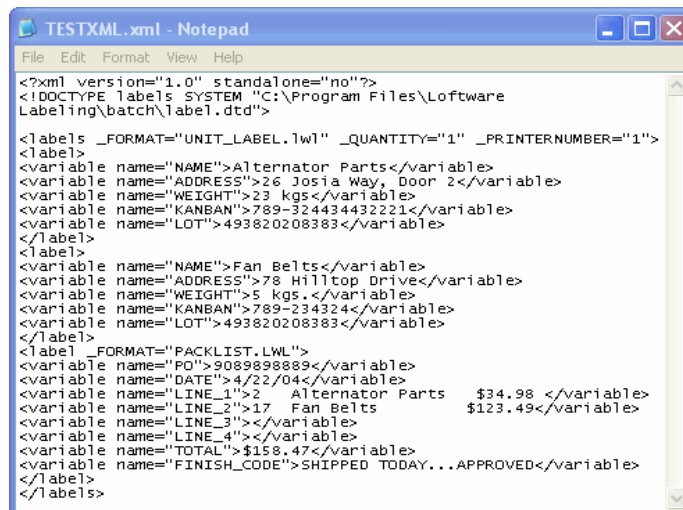


Figure 1-K: Simple XML file

In this example, one label with nametag.lwl's format is printed to Printer 1. The data for fields 'NAME' and 'COMPANY' are filled with 'John Smith' and 'ABC Company' respectively.

Note: Please note that the !DOCTYPE header line wraps to the next line in this example. Your file would have this all on one line.

Example 2: More Complex Xml File



```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "C:\Program Files\Software
Labeling\batch\label.dtd">

<labels _FORMAT="UNIT_LABEL.LW1" _QUANTITY="1" _PRINTERNUMBER="1">
<label>
<variable name="NAME">Alternator Parts</variable>
<variable name="ADDRESS">26 Josia Way, Door 2</variable>
<variable name="WEIGHT">23 kgs</variable>
<variable name="KANBAN">789-32434432221</variable>
<variable name="LOT">493820208383</variable>
</label>
<label>
<variable name="NAME">Fan Belts</variable>
<variable name="ADDRESS">78 Hilltop Drive</variable>
<variable name="WEIGHT">5 kgs.</variable>
<variable name="KANBAN">789-234324</variable>
<variable name="LOT">493820208383</variable>
</label>
<label _FORMAT="PACKLIST.LWL">
<variable name="PO">9089898889</variable>
<variable name="DATE">4/22/04</variable>
<variable name="LINE_1">2 Alternator Parts $34.98 </variable>
<variable name="LINE_2">17 Fan Belts $123.49</variable>
<variable name="LINE_3"></variable>
<variable name="LINE_4"></variable>
<variable name="TOTAL">$158.47</variable>
<variable name="FINISH_CODE">SHIPPED TODAY...APPROVED</variable>
</label>
</labels>
```

Figure 1-L: Complex XML File

This example prints 3 labels to Printer 1. The first two labels use the 'UNIT_LABEL' label format and the last one overrides the header with a different format called 'PACKLIST'. Subsequent labels will revert back to the header if the _FORMAT command is not overridden again. Some of our customers have been known to stack thousands of different label requests targeted to hundreds of different printers into a single XML file using this technique.

Note: Please note that the !DOCTYPE header line wraps to the next line in this example. Your file would have this all on one line.

Providing Data for the General Case

The Concept of "Data Push"

Although the LPS has the ability to access your data through 32-bit ODBC drivers, experience tells us that this is often the incorrect approach. Chances are that your application already has access to the data. Because of this, it is easier and faster for your application to "push" the data to the Back End (the LPS). There usually is no need to further complicate your system by having to install ODBC drivers on the LPS server and configure the labels to grab data from across the LAN or WAN. Worse, if your data is on a UNIX system, why complicate matters by having to go through bridges and connectivity software? The concept of "Data Push" is meant to simplify your system in setup, performance, and maintainability.

There are cases where you must require the LPS to "pull" the data from your database. Perhaps your application is running on an RF hand held device that does not have access to the data. Only "pull" the data when there is no other choice.

Maximizing System Flexibility, Minimizing Maintenance

The LPS system allows our customers to set their labeling systems up for the general case. By following a few very simple rules during setup, impressive flexibility can be achieved:

- **Existing Labels can be modified without changing code.** This is accomplished by writing your code to extract the variable field names from the label file before each print. If fields are removed or added, your code takes care of it “on-the-fly.” This can be done with the ActiveX and .NET programming controls for PC applications. Host applications can achieve this by using the socket interface. This technique can be circumvented by just sending all data fields in each request; the label will only access the data that pertains to it.
- **New Labels can be added without changing code.** By constraining the list of available field names in design mode, designers are only able to use variable fields that your Front End program knows how to handle.
- **New Printers can be added from different manufacturers.** When printer manufacturers and models change, simply re-save your label with the new model number. The Front-End application need not know anything about this. Depending on how different the printer is, some label design ‘tweaking’ may be needed.
- **Serialized fields, variable graphics, EPC RFID data and special check digits can be added without changing code.** Anything of a nature that is added or changed in the design mode automatically works.
- **Tables and Fields can be added to databases and used in new label formats without having to write code.** By keeping the constrained field list up to date with database table and field names, users are able to change their database and their labels without the knowledge of the Front-End program. The following sections explain how this is done.

Constraining Label Design Mode

The LPS system makes provisions for your Host or PC application to know what variable data fields are present in your label formats. This allows your application to decide what data to pass through the interface based on which format is being processed. This avoids having to “hard code” your system for specific labels. Users are able to make changes and add new labels to the system without having to involve programming staff. More information on constraining fields is found in the “Advanced Techniques” chapter of the LLM User’s Guide.

Example: ABC Corporation has a relational database on their host system named BigDatabase. It has two tables with several fields as shown below:

Customer	Product
Name	Part Number
Address	Color
City	Description
State	Weight
Zip	

Using Notepad or your favorite text editor, create the following file, and place it in the Program Files\Loftware Labeling directory. In this example, we have chosen to use the “TABLE.Field” convention for later use in an SQL statement.

Name the file `llmfield.lst` in the Software Labeling folder. If there is more than one `.lst` file, you can choose the appropriate one from the 'Advanced' tab in the Label Setup dialog box. If there is only one `.lst` file, then it is chosen by default.

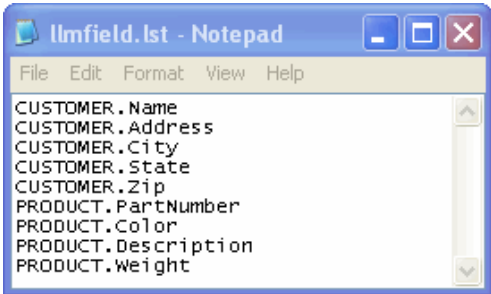


Figure 1-M: A Constrained Field List in a `.lst` file

The figure below shows how the Properties box behaves when a label is constrained with an `.lst` file. The Properties box constrains the Field Name list. In this mode, a name cannot be typed in. It must be chosen from the list, thus preventing a designer from creating a field that cannot be populated by your program.

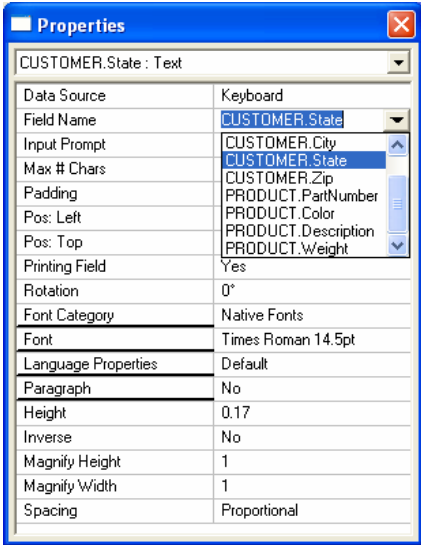


Figure 1-N: Properties box showing constrained field name list

Getting Field Names from the Label at Print Time

When the label design is saved, a file is saved with the same name with a `.TAB` extension. If the `.TAB` file does not show up in the same directory as your label format, make sure that the "Create `TAB` file on Save" option is turned on in the Design Options section of the **Options | Preferences** dialog box. For example, suppose you designed a label constrained as shown above. The name of

the label is LABEL1.LWL and contains fields for CUSTOMER.Name, PRODUCT.PartNumber and PRODUCT.Color. The label and corresponding tab file looks like the following:



Figure 1-O: Sample Label

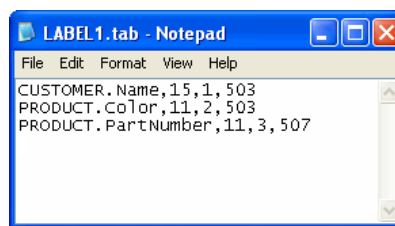


Figure 1-P: TAB file contents for sample label

Note: The .TAB file contains field name, field length, tab order, and a field type code. 503 is for variable text fields and 507 is for variable bar code fields. It is likely that the only data your application is interested in is the field name.

If your program runs on a host computer, it needs access to the .TAB file in order to know what fields are on the label. If you are using the socket interface, this information can be obtained through the socket connection rather than the tab file. If your application runs on a PC, it may use either the Loftware ActiveX control or .NET control to obtain the same information from the .LWL file. This technique is nice, because you can always be sure that the tab order and the label file are 'synced up'.

Having this information available allows your program to retrieve and supply only the minimum data necessary to print the label. If you choose to include extra data in your request, only fields in the .tab file will be accessed; the others will be ignored.

Example (continued): The ABC Corporation has written a Visual Basic Client program that determines through various criteria what labels to print. The VB program uses the Loftware ActiveX Client control to get the fields in the label and build a SQL query "on-the-fly" in order to retrieve data from the UNIX database.

Pseudo Code:

```
"SELECT CUSTOMER.Name, PRODUCT.Color, PRODUCT.PartNumber FROM BigDatabase  
WHERE CUSTOMER.Name = 'Anderson, Dana';"
```

Actual Visual Basic Code Using the ActiveX Client Control:

```
'open the database  
Set myDatabase = OpenDatabase(App.Path & "\sample.mdb")  
'build SQL statement to grab data for this label
```

```

sqlStatement = "SELECT "
ClientX1.SetLabelName "mytest.LWL"
For i = 0 To ClientX1.FieldCount - 1
    sqlStatement = sqlStatement & "[" & ClientX1.FieldName(i) & "]" & ", "
Next i
'get rid of the last comma before the FROM clause and append key
sqlStatement = Left(sqlStatement, Len(sqlStatement) - 2) & " FROM Newwar WHERE
NAME1='" & cboRecordChoice.Text & "';"
'grab the record and populate the data
'only grab the fields we need for this label from the database
Err = 0
On Error Resume Next
Set myRecordset = myDatabase.OpenRecordset(sqlStatement)
If Err <> 0 Then
    MsgBox "SQL Error #" & Err & " SQL = " & sqlStatement, vbInformation, "SQL
Error"
    Exit Sub
End If
'populate the label fields with the retrieved data
For i = 0 To ClientX1.FieldCount - 1
    thisFieldName = ClientX1.FieldName(i)
    ClientX1.SetData thisFieldName, myRecordset.Fields(thisFieldName)
Next i
myRecordset.Close
pickRandomRecord
myDatabase.Close

'print the label
ClientX1.PrintJob
This is a very nice way to retrieve only the data needed by the label. It
also allows your program to be field name independent. New labels can be
designed without having to change the Front End System! This powerful concept
is what we mean when we say that you no longer have to hard code label
information.

```

***Note:** The key to making this process work is to use actual table and field names in your .LST file that apply to your database. If a field or a table is added or deleted, you must revise your .LST file to reflect the change. This example only applies to the .NET and ActiveX controls.*

LPS Interfaces



What is an Interface? An interface is what allows a Front End Program to communicate with the Loftware Print Server. The Front End makes decisions as to what labels to print based on various criteria. Once this information is known, it is communicated through the LPS Interface to the Back End, where it is processed.

As described earlier, the LPS incorporates some very useful interfaces:

- The **File Interface** allows Front End applications to make requests to the LPS via a file Drop to a shared network drive. LPS detects this request and responds by printing the label. Any program in any language can use this interface.
- The **TCP/IP Socket Interface** allows Front End applications to have bi-directional communications directly with the LPS through a socket without the need for file transfer, or shared drives. Many aspects of the LPS configuration are available to the front end program

through the socket, such as Label List, Field List, Printer Configs, etc. Printer status and EPC data pass back are also available in this interface. This interface is for advanced programmers only and is only available in the LPS Premier product. This interface is usually used by UNIX programmers, but can be used by C++ or C# programmers as well.

- The **Direct Socket** interface is similar to the TCP/IP interface mentioned above. The difference is it is much easier to code to because it is unidirectional. Print Requests are assembled in an XML file and 'shot' through the socket. No status is available other than the fact that if you can't open the socket, the LPS probably is not running. This interface is usually used by UNIX programmers, but can be used by C++ or C# programmers as well.
- The **Connector interface** is a Java Bean used to 'abstract' the socket communication layer from Java programmers. This makes it very easy for Java applications written on other platforms to initiate print requests to the LPS
- The **ActiveX interface**, the **.NET interface**, and the **Reader Control interface** allow programming languages to send requests to the LPS from anywhere on the LAN, WAN, or Internet. Under the covers, these tools actually use a socket connection for speed and reliability. These tools are available to Windows applications. See Chapters 5 and 6 for detailed information on the ActiveX and the .Net interfaces. (The Reader Control is discussed in a separate Guide.)
- The **On Demand Print Thin Client** and the **Web Client** allow label requests to be made from any client PC on the network, or across the Internet by prompting the operator for keyboard or database key information. No programming is required to use these client programs. See Chapters 2 and 3 for detailed information on these.

The File Interface

The File Interface allows Front End applications to make requests to the LPS via a file Drop to a shared network drive. LPS detects this request and responds by printing the label. Simply put, the Front end will create pass file and write it to the scan directory. Any program on any platform can use this interface and it is the simplest to implement.

When you are first setting up the LPS, you will use the File Interface to test and verify that the system is working properly. You may then use this interface for production, or choose another interface that better suits your needs.

Simply set the scan directory as noted in the configuration applet section, create a .pas or .csv file with the appropriate data and copy the resulting file to the scan directory. Make sure you do a test print first to verify that the printer and label are functioning properly. Once you have performed these tests, you may begin dropping requests to the scan directory from your own application.

ActiveX / .NET Interface

Programmers developing their own Windows applications in 32-bit languages supporting ActiveX and/or .NET Controls can easily interface their own applications with the LPS. These controls connect to the LPS using a TCP/IP socket and effectively abstract any knowledge of the printer languages from your application. This interface uses properties to set the information needed for the label request and therefore does not follow the pass, csv, or xml data structures.

Note: *Loftware has three distinctly different controls - ActiveX Client, .NET, and the Internet ActiveX. Please see their respective chapters for information on the differences between each and how to use them. The Loftware Reader Control has recently been added to support RFID Tag reading and writing using Loftware-supported RFID readers. The Reader Control is documented in a separate guide.*

Connector Interface

The Connector was developed in response to our customers' needs for easier connectivity between their UNIX systems and the LPS. Oracle Applications can be enabled for RFID and Bar code printing by simply installing the Connector with no programming required.

Java programmers can use the Connector's Bean interface to communicate with the LPS from UNIX systems. Essentially, it 'abstracts' the socket communication layer to the LPS. This makes it very easy for Java applications written on other platforms to initiate print requests to the LPS. Refer to the Connector documentation online for more information on this powerful connectivity tool: [www.loftware.com/index.cfm/Support/User 's Guides](http://www.loftware.com/index.cfm/Support/User's_Guides)

Direct Socket Interface

The Loftware Print Server (LPS) can be configured to accept direct socket connections that do not require the use of Loftware's messaging protocol. This connection is a one-way communication that is much simpler to program than the bidirectional TCP/IP interface mentioned above. The client connects, sends data and closes the connection. There is no response from the LPS. This interface is a quick and easy solution for sending XML files to the LPS through a socket connection on UNIX systems without the need for shared drives or connectivity software.

Configure the Direct Socket Interface

The Direct Socket Interface (DSI) is off by default for security reasons. If you wish to enable it, follow these steps.

- Verify that the default scan path is configured to scan for XML files.
- Launch the *Loftware Print Server Configuration Utility* from the Control Panel.
- Highlight the default scan path (the first path listed under the *General* tab) and click *Edit*.
- In the *Scan For* section of the *Directory Setup* dialog, check *XML Files*.
- Click *OK* to accept the settings in both the *Directory Setup* dialog and the *Loftware Print Server Configuration Utility*.
- Launch the *DSIConfig.exe* application in the *Loftware Labeling* directory.
- On the LPS Direct Socket Interface Configuration dialog, check *Enable Direct Socket Interface*.
- Leave the *Listening Port* set to 2813.
- Click *OK* to accept the settings.
- If the LPS is running, it must be restarted for the changes to take effect.

Note: *The Listening Port should remain set to 2813 unless the network or a firewall blocks communication on that port in which case it may be changed to any other value within the valid port range.*

Integrating with the DSI to print XML files

- Write a client application that opens a socket connection to the LPS using a call to a Berkly compliant connect() or a WSACconnect() for Win32.
- Once the connection is established, send a properly formatted XML print job request to the LPS through the socket connection.
- Close the socket connection.

Note: The XML data must be sent all at once. Only 1 XML request can be sent per connection but it can be a stacked request. Stacked requests for multiple printers are acceptable. The socket connection MUST be closed immediately after the XML data is sent.

Direct Socket Interface Demo Application: DSISend

To further demonstrate how to integrate with the Direct Socket Interface, please see the demo application DSISend. It's written in J# and can be viewed and built in Microsoft Visual Studio .NET 2003©. Use any other editor if you simply want to view the code. DSISend is located in the following directory under Loftware Labeling: "\Sample Programs\Direct Socket Interface\J# Sample\DSISend."

TCP/IP Socket Interface

Loftware's TCP/IP Socket Interface brings a new dimension to label printing! The TCP/IP Socket Interface is a synchronous interface to the LPS that allows submission of a print job, acknowledgement of the print job, and any delays or errors related to the print job. It is much more informative than file-transfer or direct socket solutions, and allows you more control in the customization process. Due to the synchronous nature of this integration method, no other jobs may be submitted until a response is received back for a particular request. The TCP/IP Socket provides for:

- Seamless integration into the LPS.
- No File System Connectivity to host application is required.
- Faster request processing.
- Bi-directional communication between the LPS and the Front-End, meaning, Status Update Responses are sent upon the completion, failure, or delay (due to printer error) of a job request. Information about the labels, fields, and configured printers is also available to your program through this interface.
- RFID EPC data can be passed back to the calling program through this interface.
- The Loftware Connector uses the TCP/IP socket to seamlessly bridge UNIX applications to the LPS. All Loftware printing clients including the ActiveX and .NET controls also use the socket interface.
- This interface is for advanced programmers only. For more basic socket integration, refer to the Direct Socket Interface (DSI) described above.

The TCP/IP Socket was developed for those computing environments that are unable to use the ActiveX or the File Interfaces. For example, if your application resides on a UNIX or AS400 Platform on your network, but you would rather not use the FTP or shared directories approach, you could optionally communicate directly with the LPS through its socket interface. The LPS

ships with a sample program written in Berkley Compliant 'C' that shows how to do this. This program has been compiled and successfully tested on Win 2k and LINUX platforms. The program is included to give exact insight into how to harness the power of this interface's complex protocol.

For more information on the TCP/IP Socket Interface and the sample thin client program called "LPSSend," see the next section below.

We strongly suggest testing your labeling system first by manually creating files for the LPS as described in the 'Files Interface' section. Once you have validated printer connections and your labels are printing as expected, switch to using the TCP/IP Socket Interface.

LPSSend

The sample program described above is called LPSSend. It demonstrates how to connect to the LPS through a socket and comes with the complete source code. Running this sample program is the best way to learn about socket connections.

*Note: This section is for advanced users and assumes you are familiar with terminology and usage of command prompts and programming. A document called **LPSSend_ADI.doc** contains additional examples and sample code information and is found in the directory mentioned below.*

LPSSend.exe and its source files are located in the **C:\Program Files\Software Labeling\Sample Programs\LPS_Send** directory. Be sure to refer to the **LPSSend_ADI.doc** file for detailed programming information. This document also contains important information for passing data back to the calling program; like EPC RFID data.

LPSSend Files

- lpssend.cpp is the actual code and implementation of lpssend. It is coded in Berkley Compliant C even though it has a .cpp extension. If you want to view the code behind lpssend, open it in Microsoft Visual Studio or another editor.
- The three .h files are the header files for the lpssend project. They are required for building of lpssend to be successful.
- The .dsp file is the lpssend project file for use with Microsoft Visual Studio.
- The .dsw file is the lpssend workspace for use with Microsoft Visual Studio which contains the LPS .dsp protect as well as the external dependencies (.h files)
- The .exe file is the lpssend executable.

The following steps are designed to give you some insight into how LPSSend works. Much of the code in LPSSend can be re-used by your own application. Please take the time to understand how LPSSend works. After the steps are fully understood, you are able to "cut out" the parts that you need and "paste" them into your own code.

***Important Note:** Remember that the socket interface to the LPS is synchronous in nature. Your program **MUST ALWAYS WAIT** for a confirmation response back from the LPS before proceeding! This applies for Log In and Log Out as well as print requests*

Step 1. Obtain LPS Address

The first step of LPSSend is to read in the arguments or parameters from the command prompt. Once these parameters have been read in, LPSSend uses the IP address and port supplied to open a

socket connection to the LPS. Your program may hard code these values, or get them from a different place.

Step 2. Log into LPS

Once LPSSend determines that it has opened a socket connection to the LPS, a login request is sent. LPSSend sends the Computer name, User name, and version number to the LPS requesting a login. The actual protocol used to request this login is also sent. The code is: REQ_LPSLOGIN.

Step 3. Wait for Login Confirmation

After the login request is sent, LPSSend waits for a response from the LPS. The LPS sends back another protocol command to the LPSSend client: RSP_LPSLOGIN. LPSSend is successfully logged into the LPS.

Step 4. Prepare Print Request

Now that LPSSend is logged in, it prepares the Print job request. LPSSend creates a buffer large enough to store the pas, csv, or xml file specified in the command prompt parameters. The file is read into the buffer, and packaged with the printer number or printer alias, job type (pas, csv or xml), job name (if specified), as well as the protocol command: REQ_SNDJOB.

Step 5. Send Print Request, Wait for Response

When a print job has been sent to the LPS, LPSSend waits for the LPS to finish processing the job and send back a response. The LPS sends the protocol command RSP_JOBUPDATES, along with some extra data about the job. The four possible responses to a sent job are: **1.)** Critical Failures, **2.)** Printed, **3.)** Printed with Errors, and **4.)** Printer Errors. LPSSend displays this information for the user to acknowledge the status of the job sent. Your program should handle the responses in an appropriate manner. For more information on the meaning of Status responses, refer to the Status Client section of Chapter 2.

Note: Repeat Steps 4 and 5 as many times as needed without logging out.

Step 6. Logout Request

Once LPSSend's job has been sent and processed, LPSSend logs out from the LPS, with the protocol command: REQ_LOGOUT.

Step 7. Wait for Logout Confirmation

LPSSend waits for a response to the log out request from the LPS. The LPS sends back the protocol command RSP_LOGOUT.

Step 8. End

Once LPSSend has received its response to the logout request; LPSSend closes the socket connection and frees the buffer that was allocated for the print job request.

LPSSend Usage

LPSSend.exe is a sample program that contains all the documentation needed for TCP/IP connections. You can use LPSSend to learn about socket connections. Running LPSSend is a great way to test these connections, and Loftware strongly recommends that you create a simple file in order to do this.

Run LPSSend from a command window using the following information as arguments:

```
LPSSend Address -pPort [2723] -tJobType [0]
-nPrinterNumber |-aPrinterAlias -jJobName -fJobFile
(The numbers in brackets [] are default values.)
```

Options

Address	Specify the LPS IP address
-p	Specify the port (default: 2723)
-t:	JobType
-t0	Pass File (default)
-t1	CSV File
-t2	XML File
-n	Loftware Printer Number
-a	Loftware Printer Alias (Either Number or Alias is sent)
-j	JobName Name of Job
-f	Filename of job to send (with path)
-r	(By itself) Displays a chart of the return code values
-h	(By itself) Displays this help (also displayed by -?)
-l	Login type [0] default - log in only , no request made as yet
	-11 - login, send job request, wait for status
	-12 - - login, send job request, no wait for status

LPSSend Usage Examples

Example 1: `lpssend 172.16.0.64 -n1 -flabel1.pas`

This example sends the label1.pas file, located in the same directory as lpssend.exe, to the IP address 172.16.0.64 with default port of 2723 and default jobtype of 0 (.pas file) printed on Printer 1.

Example 2: `lpssend 172.16.0.64 -t1 -n2 -f "C:\Program Files\Loftware Labeling\Batch\label1.csv"`

The above example sends the label1.csv file, located in the Loftware Labeling Batch directory, to the IP address 172.16.0.64 with default port of 2723 and a jobtype of 1 (.csv file) printed on Printer 2.

LPSSend Label Example

- For this example, a label called "lwlabel.lwl" is created in Loftware Label Manager Design for a Datamax Titan 6200 Printer, configured as Printer #8.

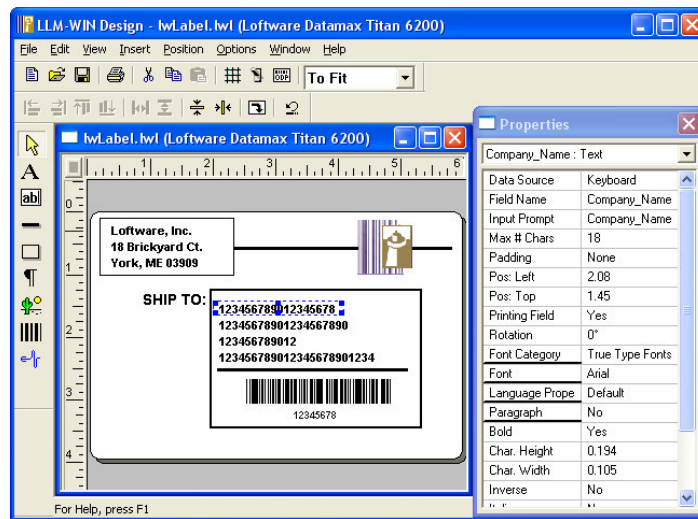


Figure 1-Q: lwlabel.lwl

- A .pas file called lwlabel.pas is created for this label, and is saved to the TCPItoLPS_Request directory. (If your program builds a .pas file in memory, do not forget the <CR> <LF> at the end of each line!)

```
*FORMAT, lwlabel.lwl
*JOBNAME, SampleJob029
*QUANTITY, 1
*PRINTERNUMBER, 8
COMPANY_NAME, Aztec Corporation
ADDRESS_1, 1221 Inca Dr
ADDRESS_2, P.O. Box 17
CITY_STATE_ZIP, Yuma, AZ. 63325
Bar Code, 552-300
*PRINTLABEL
```

Note: If the file is located in any directory other than TCPItoLPS_Request, you must provide the complete path to the directory. (Example: C:\Program Files\Software Labeling\Batch\lwlabel.pas)

The command window displays the result of the LPSSend example.

```

Command Prompt
C:\Program Files\Software Labeling\Sample Programs\TCPIPToLPS_Request>lpssend 17
2.16.0.68 -n8 -flvlabel.pas
*LPSSend: Connected to "172.16.0.68" on port 2723.
*LPSSend: Sent Log In Request.
*LPSSend: Received Log In Response from "RICA-2000".
*LPSSend: Sent Print Job Request.
*LPSSend: Received Status: "Printed" for Job "28" on Printer "8".
*LPSSend: Sent Log Out Request.
*LPSSend: Logged Out.

C:\Program Files\Software Labeling\Sample Programs\TCPIPToLPS_Request>lpssend 17
2.16.0.68 -n8 -flvlabel.pas
*LPSSend: Connected to "172.16.0.68" on port 2723.
*LPSSend: Sent Log In Request.
*LPSSend: Received Log In Response from "RICA-2000".
*LPSSend: Sent Print Job Request.
*LPSSend: Received Status: "Printed" for Job "29" on Printer "8".
*LPSSend: Sent Log Out Request.
*LPSSend: Logged Out.

```

Figure 1-R: Command Window displaying the LPSSend information

The label requested above prints out as displayed below:

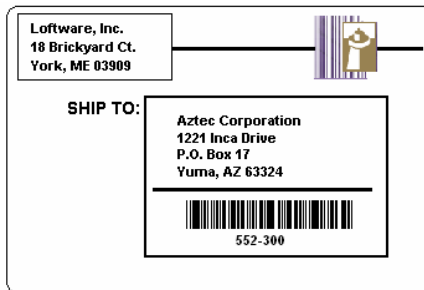


Figure 1-S: Completed Label using LPSSend

LPSSend Return Codes

LPSSend returns a code at the end of execution. The following chart explains the meaning of each of these return codes.

Return Codes for Main

100's: GENERAL MAIN EXITS

- 101: Display Return Codes
- 102: Display Args
- 103: Invalid Args
- 104: Missing Args
- 105: File Name or Path not found (of the pas, csv or xml file)

200's: SOCKET ERRORS

- 201: Error on Send Login Request
- 202: Error on Receive Login Response
- 203: Error on Send Print Job Request
- 204: Error on Receive Print Job Response
- 205: Error on Send Logout Request
- 206: Error on Receive Logout Response

300's: RECEIVED OUT OF FRAME DATA (MAGIC NUMBER)

- 301: Receive Login Response
- 302: Receive Print Job Response
- 303: Receive Logout Response

400's: FAILURES

- 401: Failed to Open Server
- 402: Login Failed
- 403: Print Job Failed
- 404: Logout Failed

500's: END OF MAIN (STATUS OF JOB)

- 501: Critical Failure
 - 502: Printed
 - 503: Printed with Errors
 - 504: Printer Error
- End of Main Returns 0 (should never happen)

Hints and Troubleshooting

Note: Many errors can be evaluated by reading the LPS entries in the Event Viewer. See the next section for information regarding these entries.

The following points help you to troubleshoot problems with the Loftware Print Server. Please go through this list before involving Loftware's technical support department.

- If you have trouble with the LPS running as a service. Stop the service and try again in Interactive mode. This mode will provide error messages that you would not see running in Service mode.
- If the LPS service is running and you have made any changes to the way the hardware key is configured, you must stop and then restart the service in order for the changes to be recognized.
- After you are finished with the implementation/debugging phase of your project, make sure that you turn off 'Event Logging' and 'Create Trace Log' in the logging section of the LPS Configuration applet. Otherwise, your event log may be flooded with messages that you do not need for production and your hard drive could be overwhelmed with trace files.
- Make sure that purge settings are set in the 'Housekeeping' tab of the LPS configuration applet. These can always be optimized as you get used to your system and its performance.

- Make sure that the LPS is not scanning the root directory of any drive. In addition, when the scan directory is set, it must already exist.
- If using pass files, make sure that there are no spaces before or after your label name on the '*Format' line.
- If you are scanning a network drive or printing to network printers, the service must be configured to start up with a specific account that has access to all the network resources that need to be used. The account must have administrative privileges.
- If you are using pass files, and you do not specifically want to print spaces at the leading or trailing end of any field, configure the LPS to trim 'Both' in the control panel configuration utility.
- If you receive an 'Error: 2140' message explaining that the service could not be started, it could mean that the user or domain \user account does not have Admin privileges on the local system.
- Do not set your scan directory to be the same as the \batch directory. The activity of creating and deleting the .tmp batch files causes LPS to needlessly check the directory.

WDPING and WDLOG Diagnostic Utilities

It is often important to be able to check the status of your printing systems to see if they are working correctly. The WDPING and WDLOG utilities allow you to do this from across the network without disturbing your labeling system in any way.

Both utilities are installed to the "Program Files\Software Labeling" subdirectory by default. When invoked from a command prompt, they display information on the screen based on the parameters supplied. Issuing the WDPING or WDLOG command with no parameters displays the usage syntax parameters.

To open the Command Prompt and use WDPing (*Example*):

1. Start > Programs > Accessories > Command Prompt
2. Type in cd program files\software labeling; press Enter.
3. Type wdping -i 172.100.0.1

The Software diagnostic utilities support the **-?** command line parameter that displays all useable parameters. Command line options do not need to be in any specific order.

Example: wdping 172.16.0.9 -u is the same as wdping -u 172.16.0.9

Note: Many of the following settings are used by Software technicians for troubleshooting purposes. End users of the LPS commonly use "Broadcast" and "Status."

WDPing parameters

Note: If, for security reasons, you do not want your server to respond to a ping from one of these utilities or a broadcast from an LPS client program, call Software Technical Support for assistance with this advanced option.

Broadcast (-b) - Broadcasts a “find LPS servers” message across the NT network and returns the ComputerName and IP Address of any Loftware servers that are currently running.

Command Syntax: **WDPing -b**

Output:

```
Server SHIPPING located at 172.100.0.1
Server RECEIVING located at 172.100.0.2
```

Status (-s) - Sends a status request to a specific LPS server identified by an IP Address. This IP address is obtained from performing a “-b” broadcast as mentioned above. If the server is running, the following is returned:

- Round trip time for the LPS response
- The computer name where the LPS resides
- The number of currently connected On-Demand Print Clients
- The High Water Mark (HWM) indicating the maximum number of clients connected at any one time since the LPS service was started.

Command Syntax: **WDPing -s 172.100.0.1**

Output:

```
Response received. Roundtrip time 19 Milliseconds.
SHIPPING has 5 connected clients. HWM is 11
```

Client Licenses (-i) – Retrieves the Client Licenses in use by the Loftware Print Server. You must include the IP address of the server. **Note:** This feature is only available in Versions 6.0 or higher of WDPing and the LPS.

Command Syntax: **WDPing -i 172.100.0.1**

Output

```
LPS Client Report
=== =====
Max Clients: 4
In Use      : 3
```

Client Information (-c) - Sends a request for Client information to a specific LPS service identified by an IP address.

Command Syntax: **WDPing -c 172.100.0.1**

Output

ID	ComputerName	User	Address	Port	Last Activity
1	SHIP_PC_1	SHIPPER1	172.100.0.100	39172	19990924 11:00:18

The ID is assigned to the Client by the LPS when the client connection is initiated. The Computer Name is the name of the PC where the client is running. User is the user currently logged onto the Client PC. Address is the IP Address of the Client, Port is the TCP port connection for the server and the client, and Last Activity is generally the time that the client last printed a label.

CPU Usage (-u) - Sends a request for CPU Usage information to a specific PC running the LPS service. If the service is running, the following information is returned:

Command Syntax: WDPing -u 172.100.0.1

Output:

```
Service start: 20030924 10:09:35

Running Time: 40 Days, 13 Hours, 21 Minutes, 07 Seconds
CPU Usage      : 000:00:00
Avg CPU Usage: 5.27%
```

Thread	CPU Time	% of CPU	% of Proc
=====	=====	=====	=====
ScanThread	000:00:00	0.00	0.00
PollThread	000:00:00	0.00	0.00
SockSvrThread	000:00:00	0.00	0.00
WorkerThread 0	000:00:00	0.00	0.00
DGramThread	000:00:00	0.00	0.00

Note: Most of the returned settings are for technician use only.

Default Response time (-w) - Used in conjunction with other WDPing command line parameters, this sets the time in seconds that the WDPing utility waits for a response from the LPS service. The default value is 2 seconds. Under normal circumstances, a response is received well within the default time. The only time you may need to increase the default value is if network traffic is heavy.

Command Syntax: WDPing -s 172.100.0.1 -w10

Thread Status (-t)

Note: This setting is for technician use only.

This sends a request for thread status to a specific LPS server identified by the IP Address supplied on the command line. If server is running, the following information is returned:

Command Syntax: WDPing -t 172.100.0.1Output:

```
ScanThread      Running
PollThread      Running
SockSvrThread   Running
WorkerThread 0  Active IDLE
DGramThread     Running
LogThread       Inactive
```

Communication Port

Note: This setting is for technician use only.

Used in conjunction with other WDPing command line parameters, this specifies the TCP/IP port that is used to communicate with the LPS. The default is 2723. This is only needed if your LPS service is running on a user specified TCP/IP port.

Do not change this setting unless instructed to do so by a certified Software technician.

Command Syntax: WDPing -s 172.100.0.1 -p3100

Output Capture (-o) - Used in conjunction with other WDPing command line parameters, this specifies a file name for capturing the data that is returned by the various WDPing requests. If no file name is specified, data is captured in "Program Files\Software Labeling\wdping.txt".

Command Syntax: **WDPing -s 172.100.0.1 -oLPS.LOG**

WDLLog parameters

Logging Data - Requests interactive logging data from a specific LPS server identified by the IP Address and outputs the data to the screen. Interactive logging data is configured in the LPS Configuration Applet. Choose one or more of the following types of logging: Log Commands, Log Batch, Log Csv, Log Pass, Job Numbers, Settings/Configuration, Checkpoint, Verbose Security, Display Files, Client Debug, Client Connections, Client Audit, or Client SQL.

Command Syntax: **Wdlog 172.100.0.1**

Communication Port

Note: This setting is for technician use only.

Same as **-p** for **WDPing**, see section above.

Command Syntax: **Wdlog 172.100.0.1 -p**

Output Capture - Same as **-o** for **WDPing**, see section above.

Command Syntax: **Wdlog 172.100.0.1 -oLPS.LOG**

Logging Data: Files

Outputs interactive logging data with the **-o** parameter to both the specified file and to the screen.

Command Syntax: **Wdlog 172.100.0.1 -oLPS.LOG -b**

Reading the Application Event Log

The following sections describe the various messages that the LPS posts to the Application Event Log. There are three categories of messages: Information, Warnings, and Errors. Knowing what these messages mean could prove to be vital in troubleshooting potentially difficult problems. Be advised that many of these messages are only visible if enabled in the 'Logging' section of the LPS Configuration Applet.

Note: The log can be brought up from the Start Menu: **Start | Programs | Administrative Tools | Event Viewer**. LPS entries will be in the 'Application' log.

Event Log Informational Messages

Message 105: The service was started.

Date and time of service start.

Message 108: The service was stopped.

Date and time of service stop.

Message 109: Found and processed command file: <name>

Message 110: Passed Batch File: <name> to Batch Print**Message 111: Passed pass file: <name> to Batch Print**

If “Log Jobs” has been enabled in the Event Logging section under the Logging tab of the LPS Configuration Applet, the above entries are added after successfully locating and processing/passing the named files. Be aware that enabling “Log Jobs” could unnecessarily ‘flood’ the system log with informational messages. Once your system is up and running, you might consider turning this setting off.

Message 112: Batch Print has issued Job Number: <number>

The Loftware Batch Print sub-system assigns job numbers to each job (file) passed to it. Enable the “Job Numbers” option in the Event Logging tab of the LPS configuration utility for this data.

Message 113: User <name> has modified configuration settings for LPS service.

This message is logged when a user modifies the settings in the LPS configuration utility.

Event Log Warning Messages**Message 120: File <name> already exists! Renamed to <name> before calling Batch Print.**

The file is renamed if a batch file (.bch) or pass file (.pas) name has been duplicated before deletion by the Loftware Batch Print sub-system.

Message 121: Cannot rename/move file <name>, file does not exist.

This is an internal error caused most likely by passing an invalid batch filename in a command file, or by a user/process deleting a batch or pass file after dropping but before processing.

Message 124: File <name> does not exist

This is an internal error commonly caused by dropping a command file before dropping the batch file (.bch). In this case, the command file is renamed to *.cme.

Message 125: Warning, Loftware Print Server is configured to run in demo mode.

There is no hardware key present. If data on your label is scrambled, the Loftware sub-system is configured to run in demo mode (normal during evaluation periods).

Event Log Errors**Message 103: The control handler could not be installed.**

An internal error during service startup, possible causes: low resources.

Message 104: The initialization process failed.

Something failed during the service initialization process (startup) follows one of the more verbose messages.

Message 114: Failed to create scan thread due to low memory resources.

The NT system could not allocate the requested resources to create a thread. Check resources, correct, reboot, and retry.

Message 115: Failed to create scan thread.

Internal error. During initialization process, NT system has issued this error. Reboot (power off) and retry.

Message 116: Failed to initialize OLE Libraries.

There is an error initializing the OLE libraries (dlls). This is usually caused by a [separate] install that incorrectly overwrites one of the OLE DLLs.

Message 117: Failed to connect to Batch Print.

The Loftware Batch Print sub-system failed to initialize. The most common reason is either

incorrect Key (there is a Loftware key on the PC, but it is NOT a Loftware Print Server key) or invalid default DCOM permissions. (See the next section titled 'NT Security Settings'.)

Message 122: Invalid directory/Cannot create for <name>, Dir <name>.

The service requires a directory 'OLEBP' to be created as a child to the scan directory. On service startup, if the directory does not exist an attempt is made to create the directory. This can fail because of invalid access rights to the path, or the scan path is incorrect to begin with (deleted after configuration). The service needs access to Loftware sub-system files. The llmwdn32.ini file has been deleted or corrupted in the WinNT directory.

Message 126: Call to Remote Batch Print generated an exception!

Internal error. Contact Loftware should this error occur.

Chapter 2 Thin Client Modules

Thin Client Overview

Loftware has developed several thin client applications in order to maximize the usefulness of the Loftware Print Server (LPS). ‘Thin client’ means that these applications run across the network with a small footprint and interact with the LPS to get their information. There are five client applications in all, each with its own purpose. Documentation for four of the clients follows, while the ActiveX Client Control is documented in Chapter 5.

Notification Agent - The Notification Agent is a service that runs on a workstation or on the same PC as the LPS. E-mail, Page, and Net Send notifications of printer errors and system status can be set up for multiple operators. Notifications are filtered by server, printer, and error type; therefore, only information of interest is sent.

LPS Status Client – Allows the viewing of LPS printing activity from anywhere on the network. A ‘tree view’ of all printers shows queued jobs, jobs with error conditions, and printer status. Any number of LPS servers can be viewed.

On-Demand Print Client - Provides users of the LPS system the ability to manually select label formats and supply data from the keyboard and/or databases to print labels. This client program has the same ‘look and feel’ as our stand-alone On Demand Print Mode, but has the advantage of being ‘thin.’

LPSSend Client - A sample project used as a tool to demonstrate how to integrate with the TCP/IP Socket Interface of the Loftware Print Server. This program opens a socket connection to the LPS, sends a print job, receives status of the job from the LPS, and then disconnects. The source code for this utility is included and is available for your use. (See Chapter 1 for examples of the LPSSend Program.)

ActiveX Client Control- Programmers developing in 32-bit languages supporting ActiveX Controls can easily interface their own applications with the LPS. Use this control when your application needs to print bar code labels or RFID smart labels. The ActiveX Client control is documented in Chapter 5.

Internet ActiveX Control- 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. 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. The iX Client control is documented in Chapter 6.

Loftware .NET Control- Like the other Loftware Client Controls, this control has a “thin” footprint. The Loftware .NET Control utilizes the Microsoft .NET© framework. This control is designed for use in .NET applications. The Loftware .NET Control is documented in Chapter 7.

Loftware Reader Control - Like the other Loftware Client Controls, this control has a “thin” footprint. Like the Loftware .NET Control, the Loftware Reader Control utilizes the Microsoft .NET© framework and is designed for use in .NET applications. The Loftware Reader Control is documented in a separate guide.

Installing and Starting Client Applications

All thin client programs are installed during the normal installation of the Loftware Print Server as described in Chapter 1 of the LLM User’s Guide. No hardware key is required on client PCs. Each client utilizes a ‘client count’ license that is tracked by the server license.

Starting Client Applications via ‘Thin Install’

Note: The client install program can be found on the CD-ROM or in the directory that contains the LPS in a subfolder called ‘LPS Client Install’. The installation for the .NET control can be found in the ‘Loftware .NET Installs’ folder. Both folders can be found in ‘Program Files\Loftware Labeling’.

1. Navigate to the network drive and directory where LPS resides using Explorer.
2. Double click ‘setup.exe’ in the ‘**LPS Client Install**’ directory (under the Loftware folder).
3. Choose the Client Applications to install when prompted to do so.
4. Access the clients using the start button ‘**Start > Programs > Loftware Labeling > Print Server > clientname**’ when the thin install has completed.

Note: The .NET control has its own installation program in the ‘Program Files\Loftware Labeling\Loftware .NET Installs’ folder. The reason for this is because if the .NET framework is not already on your system, it must be installed; this would unnecessarily increase the size of the normal client install.

Note: The Loftware Reader Control has its own installation program.

Notification Agent

The Notification Agent (herein referred to as ‘The Agent’) is a service that can be run on a client workstation or on the same PC as the LPS. E-mail, Page, and Net Send notifications of printer errors and LPS system status can be set up for multiple operators. Notifications can be filtered by server, printer, and error type so that only information of interest is sent to the operator. Please keep the following points in mind before setting up the Agent:

- A MAPI based e-mail system is required on the PC where the Agent resides if you want to use e-mail notifications. Net send notifications do not require a MAPI system.
- The Agent cannot use e-mail profiles that contain a personal message store (.pst). It is a simple matter to create a profile for the agent that does not use a personal store.

- The Agent may run on the same PC as the LPS, or as a client on a separate computer. We recommend running as a client so that you can receive a notification if there is a hardware failure on the LPS server and your printing system is down. Obviously, if the Agent were on the same PC as the LPS, it would go down as well, and notifications would not be transmitted.
- Although you can run the Agent on several PCs on your network, we recommend that you use just one. Its ability to filter specific information to specific operators eliminates the need of using more than one. Running more than one could potentially cause duplicate notifications, or, in the worst case, unpredictable behavior.
- The Agent is only available in the LPS Premier Edition.

Configuring the Notification Agent

Loftware has provided a handy configuration applet that makes it easy for you to start/stop and configure the Notification Agent. Use one of the following techniques to launch the applet.

From the Windows 'Start' button:

Start>Programs>Loftware Labeling>Print Server>LPS Configuration

From the Windows Control Panel:

1. Start>Settings>Control Panel
2. Double click the "Loftware Print Server" icon from the Control Panel screen.

Notifications Tab

Note: The 'Notifications' tab is disabled if you do not have the Premier Edition of the Loftware Print Server (LPS). This tab is the only visible tab if running the Agent as a client.

The Notification Agent must be supplied with the following in order for it to work properly. The Notification tab is where you provide the following information.

- A list of servers to be monitored
- A list of operators that are to receive notifications
- Email address and/or network computer name for 'net send'
- MAPI e-mail services pre installed on the machine where the agent runs

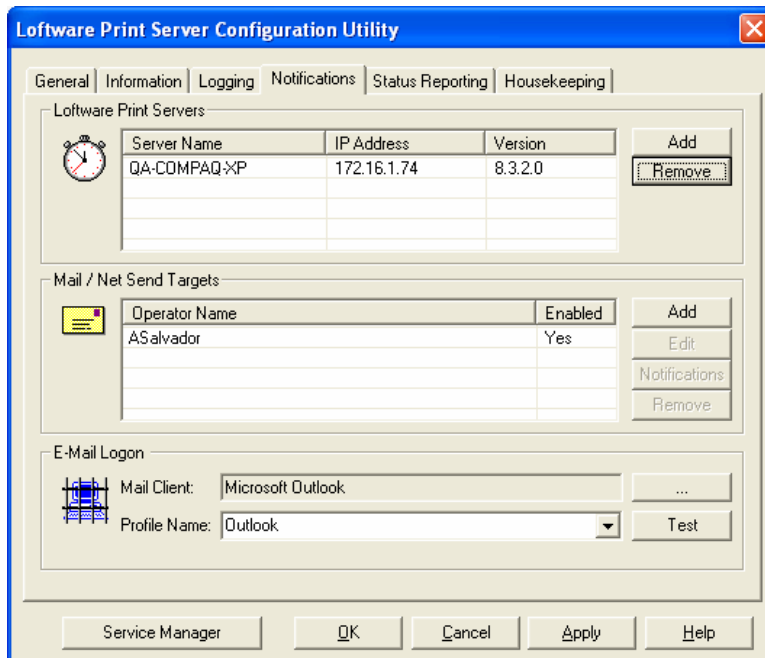


Figure 2-A: Notification Agent Configuration Utility

Loftware Print Servers - If you have an LPS running on this machine, it is automatically displayed in the server list. If your LPS is on a different machine, or you want to monitor more than one server, the 'Add' button must be pressed.

Add a Server - Figure 2-B displays the Add Server dialog box. The server selection list at the top is auto-populated with all the LPS servers that are installed and running on your local subnet. Click on the servers in the Server Selection List that you want to add (you can pick more than one); press the 'Add' button. If your server is on a different subnet, or is not listed here, you may type in its IP address and press the 'Ping' button. The Name and Version text boxes fill with the correct information if the ping was successful. If an error message is received, make sure that the IP address that you typed is correct and that the LPS at that address is running. Some LPS servers on your network may not show up in this list if they have been configured to not respond to a broadcast. All servers respond to the broadcast by default.

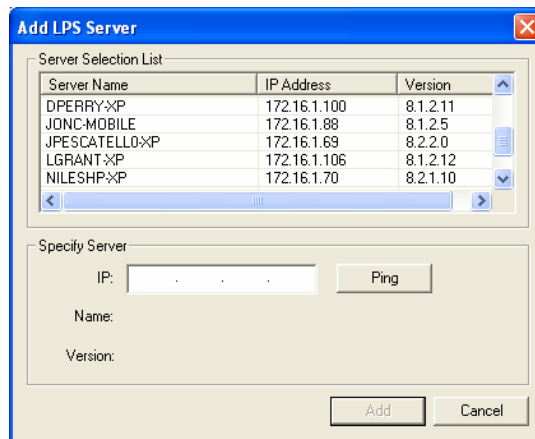


Figure 2-B: Add Server dialog box

Mail/Net Send Targets – This display grid lists all operators that you have defined in the system. There are no operators by default, so the ‘Add’ button must be pressed.

Adding a User (Operator) – Figure 2-C shows the Add Operator dialog box.

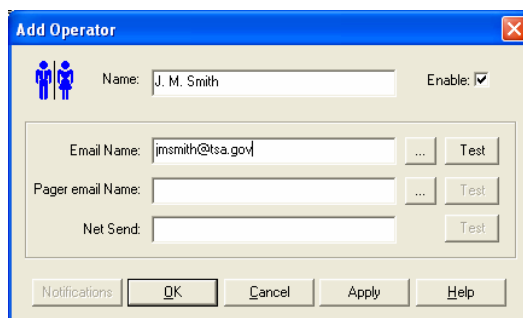


Figure 2-C: Add Operator dialog

Name: The name of the user must be provided here.

Enable: Use this to enable/disable notifications going to an operator. Perhaps they are on vacation and you want to suppress their notifications until they come back.

E-mail Name: An E-mail address or mail system name should be entered here if this user is to receive e-mail notifications. The ‘...’ button shows the -mail address book thus allowing the name to be picked from a list. The ‘Test’ button sends a test e-mail to the address provided. If the test fails:

- Be sure that you have selected a valid mail profile on the main screen of the notification tab.
- The address may be wrong; double check it.

- Your MAPI system may have a problem; try sending mail to the same address using your normal e-mail program.
- If you still have a problem, contact your network administrator.

Note: Because e-mail networks may be different, Loftware cannot help you with troubleshooting this.

Pager Email Name: The Agent does not support dial up pagers, but can send to mail enabled pagers. This prompt behaves exactly like the ‘E-mail Name’ prompt and is included in case you want notifications for a particular operator to go to both a desktop and pager.

Net Send: Net send sends the notification message across the network to the machine name supplied here. Once again, the ‘Test’ button sends a test message to verify that the machine name specified is correct. The message simply ‘pops’ up on the recipient’s screen as shown in the figure below. No MAPI system is required for net send; however, it is limited to your LAN/WAN.

Notifications Button – All notifications for all devices are sent to this operator by default. The following setup screen allows you to disable all notifications and assign particular ones to this operator.

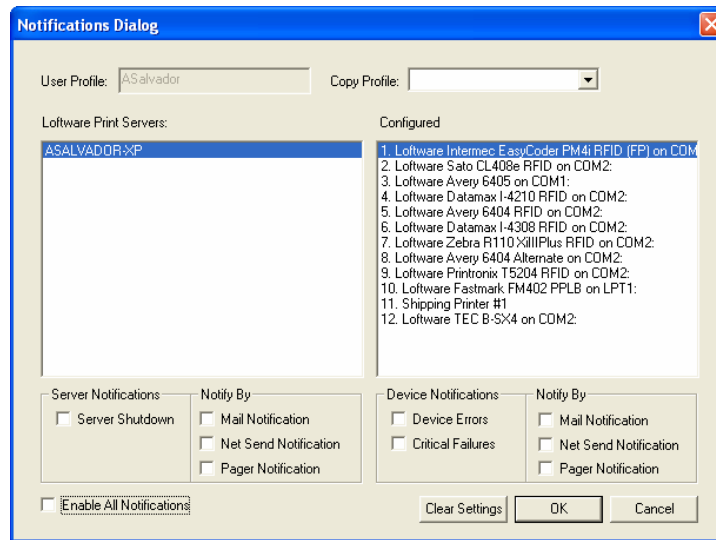


Figure 2-D: Notifications Setup dialog

When the Notifications Setup dialog is first displayed, the servers that you are monitoring are ‘grayed out’ and the ‘**Enable All Notifications**’ check box is checked. To disable all notifications, uncheck this check box, and assign errors for each printer on each server individually for this operator.

Loftware Print Servers - All LPS servers that you are monitoring are listed on the left side. Click on the server for which you wish to assign notifications. You may only select one server at a time.

Server Notifications Section - Select the ‘Server Shutdown’ check box if you wish to be notified if the server goes off line. If you have installed the Agent on a separate PC from the LPS, this setting

notifies the user if the LPS stops running or the machine it is on has a failure or unscheduled shutdown. If the agent is running on the same server as the LPS, notifications are transmitted if the service stops responding, but not if the server itself goes down.

Notify By Section - Server shut downs as well as printer errors and critical failures are transmitted by e-mail, pager, net send, or any combination of the three providing that the address for each has been set up and tested. The 'Notify By' prompt is in both the server and the printer sections because the user may want to be notified in a different way for a server shutdown, perhaps he/she wants to be paged instead. Maybe all printer notifications are sent with e-mail except for Warehouse #2 which is considered critical path. If that printer goes down, you want the operator to be paged. The way these settings work affords you maximum flexibility.

Configured Printers - After selecting a server, the 'Configured Printers' list for the selected server is displayed. There may be a short delay before it shows up. If the list does not show up, the server may be off line or not have any configured printers. Select all printers that you want to monitor on the selected server by clicking on them.

Printer Notifications and Notify By Sections- With the printers selected, choose the types of notifications that you would like to receive by clicking on the check boxes below. Printer Errors are common errors like 'Printer not turned on' or 'Printer out of labels'. Critical Failures reflect the inability of the server to submit a job because it does not understand it. Some operators may only want to receive critical errors and server shutdown errors. If you have selected more than one printer, these settings apply to all selections. If the check box is checked and 'grayed', (some printers have this setting turned on and some do not) click the check box to clear the settings for all selected printers. Click it again to turn the setting on for all selected printers.

Note: If your printer supports html status you can use your web browser to get more detailed Printer status.

See also:

Starting the Agent

Testing and Troubleshooting the Agent

Starting the Agent

Now that the Agent has been installed and configured, it is time to start it. There are two ways to do this. It can be run as a **service**, or it can be run **interactively**.

Service Agent Mode

The Notification Agent is designed from the ground up to run as a service with all applicable security. A service is an application that can be configured to run when the server is booted. It runs at a lower level (ring) than ordinary programs and can not be seen or changed by the operator.

Services can perform their functions without requiring the operator to log in, thus providing protection from intentional or accidental change (security). We recommend that you run the Notification Agent as a service while in production.

Two methods are utilized to start the Agent as a service. Each method allows you to set the service to 'Auto Start' at system boot so that you do not have to remember to start it each time your system is rebooted. The service defaults to manual start. Be sure to invoke the configuration applet before starting the service for the first time to configure it properly.

Note: Since services can not display error or status information on your desktop, starting/testing the Notification Agent in 'Interactive' mode is recommended until all notifications are working.

Starting the Notification Agent Service

Do *one* of the following:

- Navigate to the Service Control Manager, select the 'Loftware Notification Agent' and choose start or stop (Advanced Users).

or

- Launch the service by pressing the 'Service Manager' button in the Loftware Print Server Configuration Applet.
- Select the 'Loftware Notification Agent' as the service to start; press the 'Start' button.

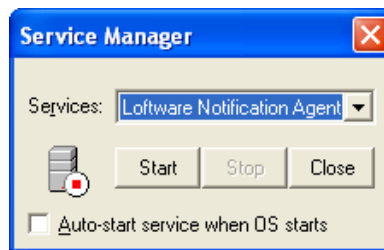


Figure 2-E: Loftware Service Manager launched from the Configuration Applet.

Interactive Agent Mode

Interactive mode runs the Agent as a normal program and is for system debugging purposes only. There is no protection from the user shutting it down, in which case notifications would stop. It is generally best to use the Agent in service mode when in production.

The **Start Menu** is used to run the Notification Agent interactively. When running in interactive mode, its status window allows you to monitor what it is doing at any given moment.

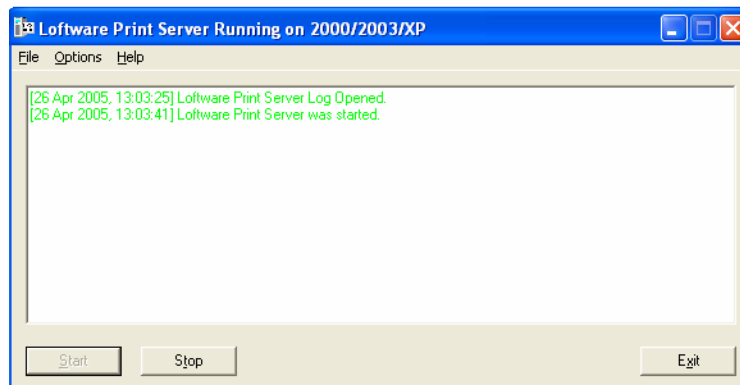


Figure 2-F: Notification Agent running interactively

Note: If you are running the Notification Agent as a service, you must stop the service before invoking it interactively.

- Use the LPS Configuration Applet to configure the Notification Agent as described in the previous section.
- The Notification Agent can be launched using the ‘Start’ button in the lower left corner of your screen. **Start > Programs > Loftware Labeling > Print Server > Notification Agent Interactive.**
- The Agent can only be started in interactive mode by using the Start button.

Testing and Troubleshooting the Notification Agent

Testing the Agent

After setting up the notifications for one or more operators, some simple tests should be done to verify that the Agent is functioning properly. Set yourself up to be e-mailed or ‘net send’ for all shutdowns, printer errors, and critical errors for all printers (default). Make sure that the LPS and the Agent are started and try one or more of the following to cause a notification transmission:

Note: Allow the Agent 30 seconds to connect to the LPS.

1. Shut the LPS down. A notification should go out if you have selected the ‘Server Shutdown’ notification (or by default).
2. Create a PAS, CSV, or XML file with an intentional syntax error in it and place it in one of the scan directories. This should generate a critical failure notification.
3. Try turning off printers, unplugging their cables, etc. This should generate a printer error notification when the next print request is made.
4. Some printers generate an error if they are out of labels/or ribbon; others do not. It is instructive to see whether notifications go out under these conditions for your particular printer. (Chapter 2 and Appendix B of the LLM User’s Guide have information on printer error messages.)

Note: Never shut printers off during media error conditions, as the buffer may contain unprinted labels!

If you get an E-mail Notification Warning (see below) when testing or configuring the Agent, the next section discusses adding a new mail account.

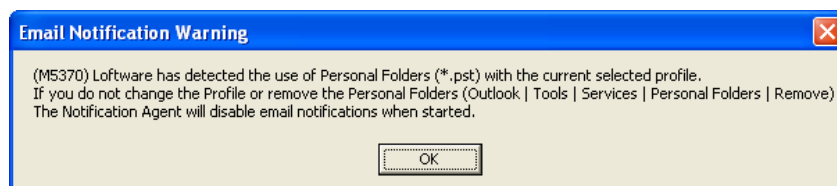


Figure 2-G: E-mail Notification Warning Message

Adding a New Mail Account for use with the Agent

To add a Mail account that does not use a .pst file (Personal Folder), first ask your Network Administrator to have a new mailbox configured on the Server, then:

1. Click Start>Settings>Control Panel; double-click on Mail.

The Mail Settings dialog box is displayed if you have added profiles previously:

2. Choose **Show Profiles**.
3. Click **Add**.
4. Choose the Information Service(s) you want to use.
5. Type in a Profile Name (the one that the Administrator created); click **Next**.
6. Type in the Name and Mailbox information; click **Next**.

Ask your Administrator for the name and mailbox information, if needed.

7. Answer the question as to whether or not you travel; click **Next**.
8. The dialog box displays "Done." Click **Finish**.

To confirm that the new mail account does not have personal folders, (.pst files) click on the new account, choose Properties. "Personal Folders" should not be displayed in the list. You may also check the mail account by going in to the LPS Configuration Utility and click on the Notification tab. If the previous warning message is not displayed, then your e-mail for the Agent is configured correctly. You may test the new mail account for the Agent and add the user by following the instructions outlined previously for the Configuration Utility Applet.

Note: This new mail account does not interfere in any way with your usual mail account with personal folders. Both co-exist and operate on the same system with no problems. The actual dialog boxes and menus may appear different depending on the operating system.

Adding a New Mail Account with Lotus Notes / Outlook 2000

1. Install Microsoft Outlook 2000.
During mail setup, choose to connect to third party e-mail programs.
2. Install Lotus Notes Client (Version 5.02 recommended).
3. Click "Show Profiles" from the Control Panel Mail icon.
4. Click ADD a new profile; select **Lotus Notes Mail**.
5. Type in LPS Agent for the Profile Name; click **Next**.
6. Enter your Lotus Notes password.
A "Congratulations" dialog box is displayed.
7. Click **Next**. The "Done" dialog box is displayed with the installed services; click **Finish**.
8. Remove any profiles not being used; choose LPS Agent as the default profile for Outlook.

Note: Outlook must be open in order for the LPS Agent to utilize the MAPI.

9. Configure the LPS Notification Agent.
10. Start the LPS Notification Agent interactively to verify MAPI logon.

Software Notification Agent with Microsoft Outlook 2002

Problem:

When attempting to run Software Notification Agent (LNA) as a service with Outlook 2002 as the mail client, e-mail notifications are not sent.

or...

When running LNA as a stand-alone executable, you receive the following warning message:

“A program is trying to access e-mail addresses you have stored in Outlook. Do you want to allow this? If this is unexpected, it may be a virus and you should choose ‘No.’”

If you click **Yes**, you receive the following message:

“A program is trying to automatically send e-mail on your behalf. Do you want to allow this?”
If this is unexpected, it may be a virus and you should choose ‘No’.

Solution:

There are new security features installed by default with Microsoft Outlook 2002, which contain locked-down settings established by the Outlook security template. In order to run the Software Notification Agent with Outlook 2002, you need to change the default Outlook security settings for the account configured to send notifications.

To enable custom security settings, your clients must be using Outlook with Microsoft Exchange Server and have either the Mailbox (MDB) or Offline folders (OST) as the default e-mail delivery location. You cannot modify the settings if a client is using a local PST file for a mailbox.

Steps to Change the Default Security Settings:

Step 1 - Install the Outlook Security Features Administrative Package

1. Run **Admpack.exe** from the \Files\PFfiles\ORKTools\ORK10\Tools\Admpack\ folder on the Office Resource Kit CD.
2. If you are installing the Outlook Security Features Administrative Package from an Office Premier Edition CD, the path is \ORK\Files\PFfiles\ORKTools\ORK10\Tools\Admpack\.
3. This executable copies four administrative files to a location you specify on your computer. You can also download the file from Microsoft’s Website.

Note: For more details on installation issues concerning the security admin package, view the readme.doc file installed with it. The following steps are also detailed in the readme file but are modified specifically for configuration with the LNA.

Step 2 - Create a public folder for the security settings

1. Before you begin to modify the security settings, you must create a public folder named "Outlook Security Settings" or "Outlook 10 Security Settings" on Exchange Server. The administrator must create this folder, using that exact name, in the root folder of the Public Folder tree. You must set the folder Access Control Lists (ACLs) so all users can read all items in the folder. However, only those users who you want to create or change security settings should have permission to create, edit, or delete items in the folder.
2. If you want multiple users to be able to edit or create items, and if the list of users can change at anytime, then you must create a security group that includes all users who you want to be able to create or change security settings. This security group should have owner permissions on the security folder. After you create the folder, you can use the template to make the changes you need.

Step 3 - Install and register the Trusted Code control

1. Copy the file Hashctl.dll from the working directory where you installed the Outlook security tools to the \winnt\system32 folder on the PC running LNA.
2. If your operating system is installed in a directory other than \WINNT, substitute the appropriate path name.
3. From the Start menu, choose Run, then type the following command line in the box to register the control:
4. `regsvr32 hashctl.dll`
5. Copy the file Comdlg32.ocx from your working directory to the \winnt\system32 folder on the PC running LNA.
6. Substitute the appropriate path name if your operating system is installed in a directory other than WINNT.
7. Choose Run from the Start menu, then type the following command line in the box to register the control:
`regsvr32 comdlg32.ocx`

Step 4 - Modify the default security settings for OutlookSecurity.oft

1. On the computer where LNA operates, launch Outlook using the account that the Loftware Notification Agent uses to send E-mail notification.
2. Open **OutlookSecurity.oft** from the working directory where you installed the Outlook security tools.
3. When asked to select a folder, select the **Outlook Security Settings** or Outlook 10 Security Settings public folder that you created on Exchange Server. The template then opens in compose mode.
4. On the Tools menu of the template, point to Forms, and then click **Publish Form**. (The folder selected should be your current folder, Outlook Security Settings or Outlook 10 Security Settings.)
5. Define a Display Name and in the Form Name box, type **Outlook Security Form**.

6. Click the **Publish** button to publish the security template in the Security Settings folder.
7. Close the Outlook Security template.
(Do not save when prompted to save while closing the template.)
8. Switch to Microsoft Outlook, click the drop-down arrow next to the New button on the toolbar, and select the **Choose Form** command from the list.
9. Navigate to the template you just created in the previous steps then select the new template name and click the **Open** button.
10. Create either a default security setting or custom settings for a specific set of users. You want to be sure that at least the mail account designated to send notifications with LNA is added here.
11. Click **Default Security Settings for All Users** to create a default security setting that is to be used by all users.
12. To create custom security settings for a specific set of users, click **Security Settings for Exception Group**, and then type a name in the Security Group Name box that describes the group. In the Members box, type the name of each user who must have custom security settings. Again, ensure that the account used with LNA is added here. Adding users from the Contact Address Book is **not** supported.
13. Go to the **Programmatic Settings** tab. Change “When Sending Items via Simple MAPI” and “When resolving names via Simple MAPI” to automatically approve. Click **Close** then **Save**.

On-Demand Print Client

***Note:** The On-Demand Print Client application is licensed by "seats," not "concurrency." This means that the first workstations to log in are recorded as a client license taken by the LPS. Once the maximum number of licenses is used, no other On-Demand client modules are able to execute, even if one logs out. If you want different workstations to be able to use client modules, you must stop the LPS Service and then restart it. This clears the license list kept by the LPS. For complete information on licensing for all Loftware Products, refer to the Loftware Website.*

The Thin Client On-Demand Print Module, herein referred to as the “Client,” provides users of the LPS system the ability to manually select labels and supply data from the keyboard and/or databases to print labels. The Web Client module has the same functionality but can be used to print labels from a remote location over the internet.

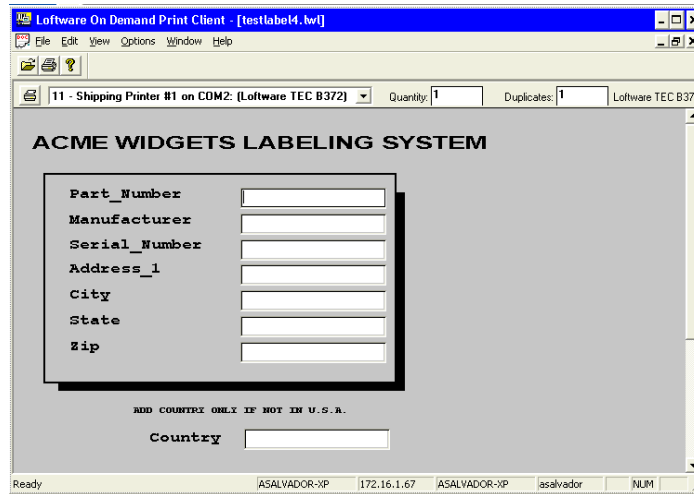


Figure 2-H: Typical Thin ODP Client Screen

Thin Client vs. On-Demand Print Mode

[Chapter 9 of the LLM User's Guide](#) describes the traditional On-Demand Print mode that our customers have used for years to print bar code labels “on demand.” The Client has the same basic functionality as On-Demand Print with the following important differences.

- √ No hardware license key is required on the Client PC. The one and only license key is located on the same PC on which the LPS are installed. This key tracks how many printers and clients are allowed to use the system.
- √ The Client requires that the LPS be installed somewhere on the LAN or WAN.
- √ Although you may have several workstations using the Client, the Loftware software is only installed on one PC and is therefore far easier to maintain.

If you use one or two “print only” stations, it is probably more efficient to use the Loftware Label Manager Print Only or Loftware Label Manager Full License. If you have three or more print stations consider using the Client in conjunction with the LPS. Also, consider using the Client if you are already using the LPS System for other purposes and the Client needs to access the same labels and printers already being used by the LPS.

Note: The Client only works if you have the LPS configured. See Chapter 1 for information on installing and configuring the LPS.

Preparing to Use the On-Demand Print Client

Note: It is assumed that you have already performed a ‘thin install’ of the client. See the previous section called ‘Installing and Starting Client Applications’.

1. Connect your printers and configure them from label design mode on the PC where LPS is installed. The LLM User's Guide has extensive information on configuring printers.
2. Design your labels, making sure that you customize your operator input screen. See the LLM User's Guide, if needed.
3. "Test print" your labels from design mode before trying to use the On-Demand Print Client.
4. Print your labels using the On-Demand Client.

If your labels do not print, try the following:

- Make sure that you are able to "test print" your labels using the Client from the server before trying to use the Client on a workstation. See the shortcut method described earlier if you do not know how to do this.
- Stop and start the Loftware Print Server and try again.
- Use the llmwclnt.ini file (documented below) to make any customizations for advanced users.

How the ODP Client Works

When the ODP Client is launched, it sends a broadcast looking for all LPS servers on your network (local subnet). When the client receives the response, it connects to the server automatically. If no servers are found, the launch fails. If more than one server is found, an arbitration dialog box is displayed prompting you to choose the server you wish to use. If your LPS server is on a different subnet, or if you wish to have access to multiple LPS servers, create an llmwclnt.ini file. See the next section for more information on the llmwclnt.ini file.

Note: If, for security reasons, you want to suppress a server on the arbitration list, call Loftware Technical Support at 603-766-3630 (follow the phone prompt for Tech Support) for assistance with this advanced option.

The Client needs access to the following LPS subdirectories: The first, called "labels" is where the .lwl file is obtained in order to prompt the operator. The second, called "layouts" is the directory where the layout files are stored. The third, called "drop" is the default LPS scan directory where the print requests are directed. If images and serial numbers are being used, access is required to these directories as well.

By default, a global share is created during the install of the LPS called 'LOFTWARE\$'. All directories mentioned above can be found under this share. If you have changed the location of any of these directories, the client must have privileges granted to it in the form of a share. See Knowledge Base Article #46413 on the Loftware Website for more information on share points.

Note: You may choose to limit the 'LOFTWARE\$' share to users of the client programs. By default, it is a global share, which can be dangerous. Use Windows Explorer to accomplish this.

The llmwclnt.ini File

In most cases, this file is not utilized. The directories referred to in the previous section are the defaults, so a llmwclnt.ini file does not need to be created unless the client PCs are located on a separate subnet from the LPS Server.

The .ini file should be named **llmwcInt.ini** and should be in the same directory as the Client. When LPS is installed, it installs a sample file called llmwcInt.sav to the same directory as the LPS. Renaming this file with an .ini extension gives you a head start in creating the file.

The following is a syntax example of what a llmwcInt.ini file looks like if you have LPS installed on two servers.

```
[Config]
;ForceSelectPrinter=1
;iniRedir = \\server1\Software Labeling
[Receiving1]
Name=JANAA
Alias=Jana Computer
Address=172.16.0.8
[DemoRoom]
Name=TRAINING
Alias=Training Room Server
Address=172.16.0.9
```

[Config] - General Client Configuration settings section.

ForceSelectPrinter - Forces the client to choose a printer based on the selected label to prevent accidental misrouting of print jobs.

IniRedir – If you wish to locate the client executable on individual machines to reduce load times across your network, you can create an llmwcInt.ini file on each machine with the iniredir section pointing back to a full .ini file that is centralized. This way, if anything in the file needs to be changed, you still have central control. If you have performed a ‘thin install’, the llmwcInt.sav file is created on the client PC for you.

[Section] – Section names must be unique and surrounded by [] s. Each section name, except config, must represent a unique name of an individual LPS print server.

Name – The computer name on the server.

Alias – The common name for the server.

Address – The IP address of the PC running the LPS. The system automatically receives this address using a broadcast technique if the address entry is not there, unless it is on a different subnet, in which case it is not found.

LabelsPath – The UNC path to the labels directory if different from the default. (ClientX)

LayoutPath – The UNC path to the layout directory if different from the default. (ClientX)

PrinterPath – The UNC path to the LPS directory. (ClientX)

ScanPath – The UNC path to the scan directory if different from the default. (ClientX)

Operation of the On-Demand Client

1. Launch the Client from a shortcut or the ‘Start’ button as described previously.
The Software On-Demand Print Client window is displayed.
2. From the On-Demand Print Client File menu, select **Open**. Choose the label file that you want to print from the Label directory.

3. Enter label information using parameters as shown in the following figure:

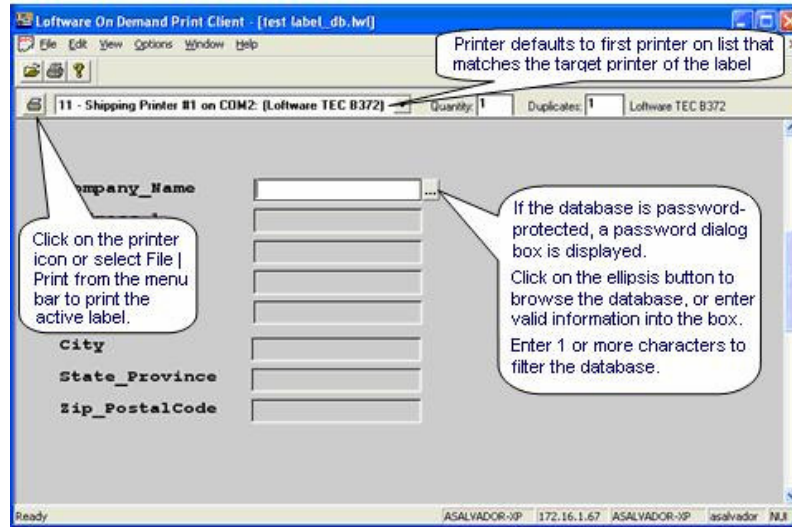



Figure 2-1: Thin Client On-Demand Print Data Entry Screen

Using the ODP Thin Client with a Database

In the sample label above, all areas are grayed out except the Database Key Field. Use one of the following options to access the database information for your label:

1. Click on the  button; the first page of records in the database is displayed.
2. Type character(s) in the **Search Key** box. When a match is found, it is highlighted.

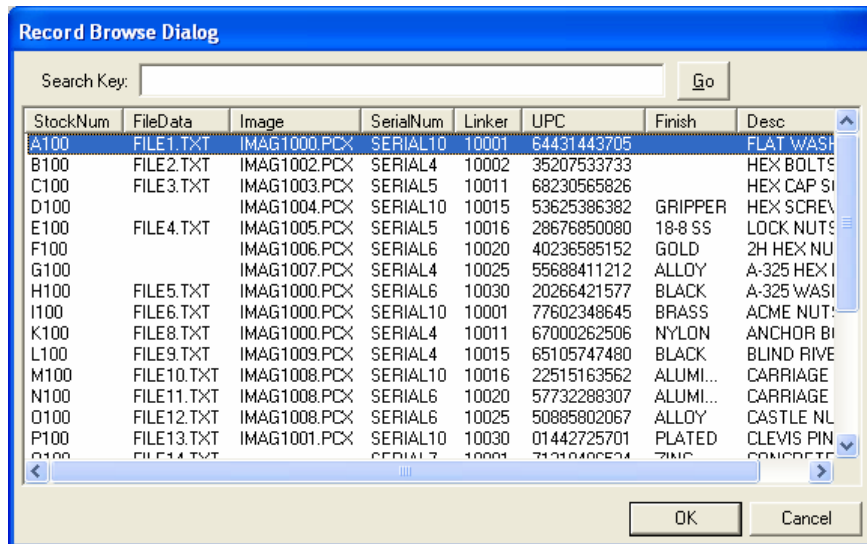


Figure 2-J: Record Browse dialog box

- Alternatively, to filter the database, enter one or more valid character(s) in the key field and click on the ellipsis button.

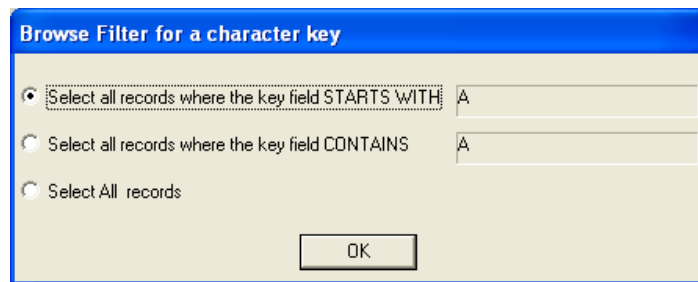


Figure 2-K: Browse Filter

- Choose one of the three filter options; the database is “filtered” to display only database fields that start with, contains specified character(s), or all records.

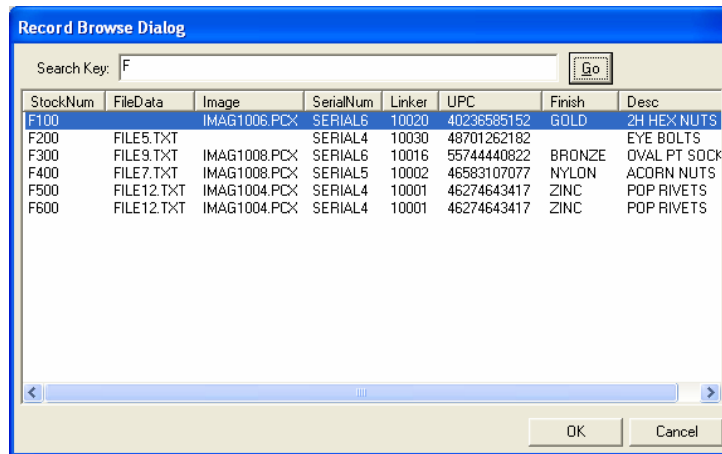


Figure 2-L: Browsing the database using the Search Key

4. Double-click the record you wish to open.
5. Print the label(s) by clicking on one of the Printer Icons, select **File | Print** from the menu, or choose **F9** on the keyboard.

Creating an Icon to Run the Thin On-Demand Print Client

Creating an icon that is attached to a particular label on a desktop provides instant printing capability! When clicked, the icon opens the ODP Client, the label requested and prints it.

To create the icon:

1. From the Software Labeling directory in Windows Explorer, locate and right click on **LLMWCInt.exe**.
2. Choose **Create Shortcut** from the menu.
3. Click and drag this shortcut to your desktop; close Explorer.
4. Right-click on the desktop icon, choose **Properties**. This displays the LLMWCInt.exe properties box.:
5. Add the name of the label that is to be printed at the end of the Target line after **.....exe**.

For example: "C:\Program Files\Software Labeling\LLMWCInt.exe" **db_1.lwl**

Where the path to the label is a UNC path to the LPS Server plus the Software share name, the example looks like:

"C:\Program Files\Software Labeling\LLMWCInt.exe" **-L"\\lps-server
hostname\SOFTWARE\$\labels\db_1.lwl"**

To specify a printer or quantity:

If you want to automatically select a printer on a per user basis, add the `-p` command to the shortcut. The format is as follows:

-Pn (Where n is any valid, configured Loftware Printer number)

The same goes for the default quantity

-Dn (Where n is a valid, non-negative default quantity)

***Note:** If the `-p` command is used and you open a label that is not designed for the printer specified, the `-p` command is ignored. For example, the client is executed with `-P4` and Printer 4 is an Intermec EasyCoder 3440, the user opens “acme.lwl” which is designed for a Monarch 9830, the default printer of 4 is ignored since the 3440 is not displayed in the printer combo on the client (remember the only printer(s) displayed match the printer(s) the label is designed for in the Client).*

Troubleshooting the Thin On-Demand Print Client

For information on your Thin Client session, click on **View | Diagnostics** from the menu bar:

The Loftware Client Diagnostics dialog box is displayed.

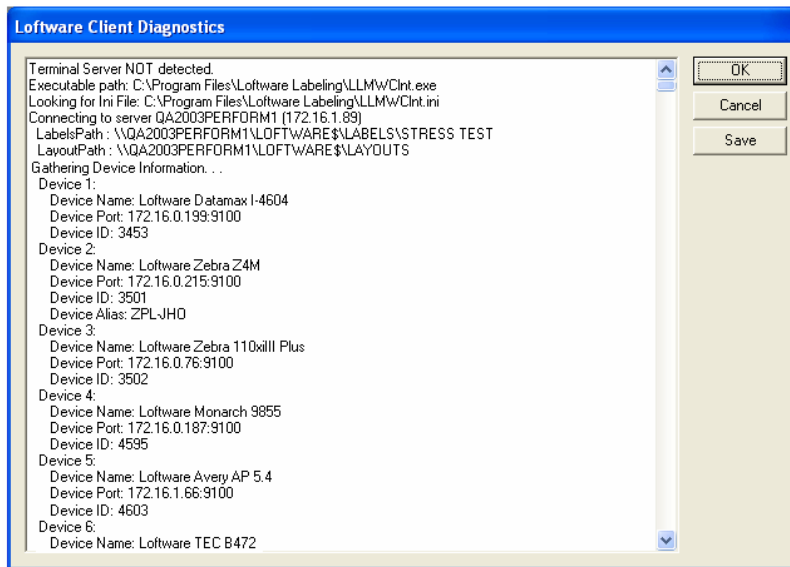


Figure 2-M: Loftware Client Diagnostics dialog box

The Client Diagnostics box gives you information on the Labels, Layout and Scan Paths, as well as printer information. If you are still having trouble printing using the Thin Client On-Demand Printing module, read the following checklist before calling Technical Support.

1. Is LPS running (service started or scanning in interactive mode)?

2. Verify that the paths you have chosen for your labels, layout and the scan path are valid. If they are not, manually set the paths in the UNC paths section.
3. Make sure that you have configured the right printer for the label.
4. Verify that all server information is correct, such as the IP address, etc.
5. Make sure that you can still test print the label from the label designer back on the server.

Beginning in Loftware Version 6.0, it is possible to configure printers with the Loftware Print Server running. The LPS monitors printer changes, and when it detects a change, the printer list is updated, and notifications are sent to the connected On-Demand Print Client(s).

Be aware that a problem could develop on the client-side if a printer is deleted on the server side, for example, an ODP Client attempting to print to the deleted printer would receive a display that states "No Printer Configured." Also, when the list of printers is updated because of a configuration change, if more than one printer of the same make and model printer exists, the default is to the first printer in the list that matches the loaded label.

Example: Say there are three Zebra 170XiIII printers configured on Printers 1, 5 and 7, the user is printing to Printer 5, and the printer configuration is changed. The default printer then becomes Printer 1. This may create a problem for a user who is attempting to print, and does not notice that the printer list has changed. The label that was previously printing without any problems seems not to be able to print. This is another case in point where a Printer Alias can be very helpful, as the client user may be more apt to notice a change in the alias of the printer, as opposed to just a number.

Status Client Application

The LPS Status Client Application allows the viewing of LPS printing activity from anywhere on the network. It allows you to get a quick 'snapshot' of printing activity throughout the shop floor. It also allows you to delete pending jobs and reprint jobs on an as needed basis. 'Tree' and 'Context' views are provided to maximize the information that you can obtain. Any number of LPS servers can be monitored. Jobs can be viewed, resubmitted, or deleted with a simple right-click of the mouse.

Note: If, for security reasons, you want to suppress a server on the LPS Server list, call Loftware Technical Support at 603-766-3630 (follow the phone prompt for Tech Support) for assistance with this advanced option.

Setup Information

- Refer to the 'Installing and Starting Client Applications' for information on how to launch the status program from either a shortcut or via the Start button after performing a 'thin install'.
- This application uses much of the same technology as the 'On-Demand Print Client'. Refer to the 'How it Works' and 'llmwcInt.ini' sections of the On-Demand Client section to review this information.

Note: If you do not have the Premier Edition, Status is only run on the LPS server.

See also:

Understanding the Screen Layout

Status Client Examples

Status Menu Choices

Operating the Status Client

The Status Client application can be launched from a shortcut or the 'Start' button. At startup, it broadcasts to all the LPS servers on your local subnet. If it finds more than one, the following dialog box is displayed: (Multiple servers may be selected for monitoring.)

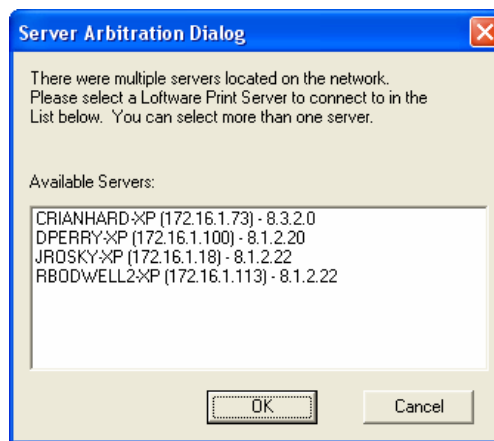


Figure 2-N: Server Arbitration dialog box when more than one server is detected

The Status Client application displays a server/printer 'tree' on the left side and a context view on the right side. It is called a context view because its display is dependent upon which 'branch' of the tree is selected.

Each server has a 'General Queue' where error conditions are logged when the system cannot figure out which printer is to receive the request. For example, suppose you dropped a file that had a syntax error that caused the LPS to not be able to parse the destination printer. The error would go into the general Queue. This queue is most helpful for reviewing mistakes that you have made during the development process. Once your requests are coming through with no syntax errors, the General Queue should not be needed.

Understanding the Screen Layout

Real Time Refreshing: All limbs of the selected (highlighted) branch of the tree are updated 'real time.' There may be a slight delay depending on how busy the server is. You can make all printer status for a particular server update in real time by selecting the server. The server is selected by default when entering the program, unless more than one server is specified.

Refreshing at Polling Interval: Any limbs that are not part of the selected branch are updated at the polling interval, which is 30 seconds by default. Shortening the poll interval speeds the screen

update, but burdens the server more. Software recommends that you leave the poll interval at 30 seconds and use the real time highlighting method described above.

Purge Level: Jobs in the context view remain until the next purge interval. See the 'Housekeeping Tab' section of the LPS chapter for detailed information on file purging.

Jobs marked as Printed: When a job in the context view is marked as 'Printed,' the LPS has finished processing this job and it remains on the system until the next purge interval. This type of job can be resent to the printer as described in one of the following examples:

Jobs marked as Pending: If the printer is in a busy or error state, new jobs that are requested of it are queued up and marked as 'Pending.' Pending jobs can be deleted by 'right clicking' on the job and choosing 'Delete' from the context menu. Pending job files allow the LPS system to recover from critical errors and are not deleted during the purge process.

Keep the following points in mind when deleting a Pending job:

- Depending on how busy the server is, there can be a considerable pause before the job is removed.
- When you delete a Pending Job, the printer goes into an error state temporarily. Notifications go out if you are using the Agent. Normal printer operations resume after the next print request to that printer.

Jobs marked as Failed: If the LPS does not understand the request that has been made, it is marked as 'Failed.' This occurs when the request contains a syntax error or invalid information. Failed jobs usually occur when you first start debugging your system. They should become rare or non-existent when your system is fully debugged. You can "right click" on a failed job and choose 'View File.' If you see a syntax error, you may correct it and hit resend.

Jobs marked as Spooled: These are jobs that have been processed by the LPS, and are being sent to the Web Listener for processing. Once the job request has been sent back to the LPS by the Listener, the context view for that job is displayed as "Requested." Once the job has been printed or fails, it moves to the Printed or Failed directory. Spooled jobs can be deleted by 'right clicking' on the job and choosing 'Delete' from the context menu.

Green icons on tree branches: A green icon indicates that the printer is accepting jobs and no errors have been detected. This is not 100% accurate and we suggest you familiarize yourself with the 'Printer Status Responses' section of Chapter 2 in the LLM User's Guide and the [Printer Error Message section](#) of the specific Printer Family in Appendix B in the LLM User's Guide.

Red icons on tree branches: A red icon indicates that a printer is in an error state. No new jobs are printed on this printer until the error is resolved. The icon at the server level goes red if any of its printers (branches) have a problem. This does not mean that the server is down; it simply is indicating that one of its printers needs attention.

Yellow icons on tree branches: Indicates that there was a Failed Job on that branch. The reason it is yellow is that although there was a failed job, new jobs that do not have syntax errors are still able to print. Like the red icon, the server icon goes yellow if one of its printers goes yellow.

As you can see from the discussion above, this program can provide a wealth of information. Hundreds of printers being driven on multiple servers can be monitored simultaneously. The best way to understand the information displayed in the Status application is by example.

Status Client Examples

Status Example 1 – Viewing Jobs

The figure below shows an LPS server.

- There are 6 configured devices on this server.
- At this moment in time, devices 1 to 5 are idle. Device 6 has 12 jobs pending.
- When the server root is selected at the top of the tree, the context view shows **all** pending and printed jobs for all printers on the selected server real time (as opposed to just the 1 printer).
- If a branch is been expanded showing **Printed/Pending/Spooled/Failed jobs**, the branch refreshes ‘real time’, and all other branches refresh at the polling interval.

Read the section on Context Menus to see how to view a file, resend jobs, and delete jobs.

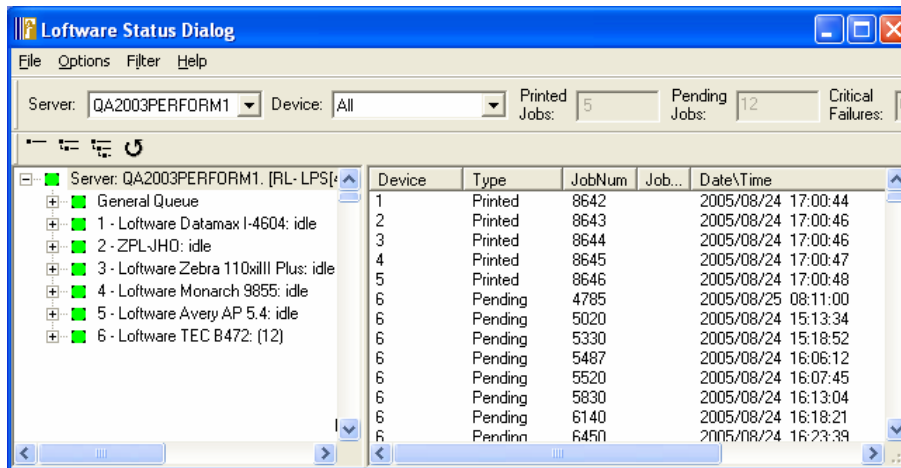


Figure 2-0: Status Client showing one server and status of jobs

Status Example 2 – Context Menus

The next figure shows a view of an LPS server:

- The number next to the name is the serial number of the LPS running on this server.
- The count in parenthesis next to the printers reflects the number of pending jobs for that particular printer.
- The context menu shown in this figure is displayed when an item in the context view is right clicked. The choices in this menu are documented in the next section, ‘Menu Choices’:
- The context view is for Device 4.

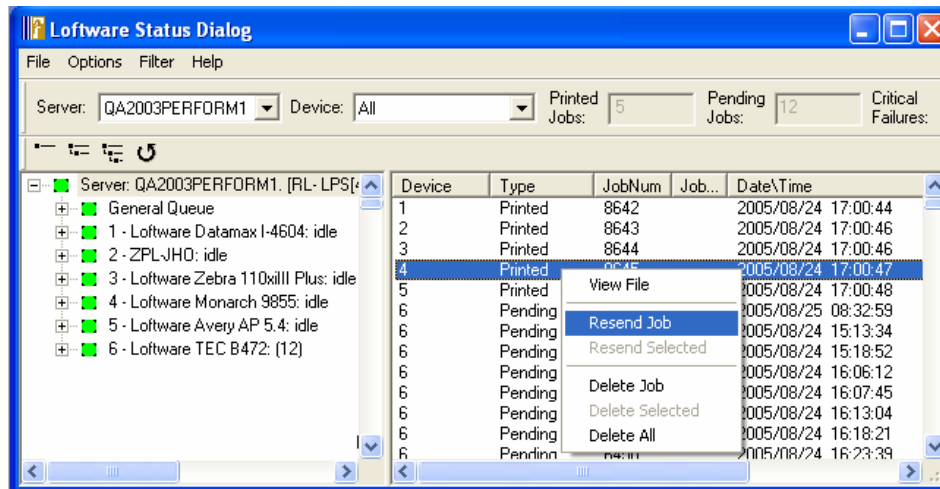


Figure 2-P: Status Client showing 6 printers and the context menu for one of the printed jobs

Status Example 3 – Monitoring Multiple Servers

The following figure shows the Status dialog connected to multiple servers.

- Five LPS servers are shown in the tree.
- Server 'TRAIN' is expanded to show its General Queue and 3 idle printers. The right column displays all the Printed, Pending and Spooled jobs for that server.
- The LPS on the server named 'RICA-2000' is not running and is therefore marked with a red icon and an 'offline' message. This could also occur if the server was extremely busy or has hundreds of printers that need to be refreshed. You may have to wait up to 30 seconds before the branch 'goes green'.

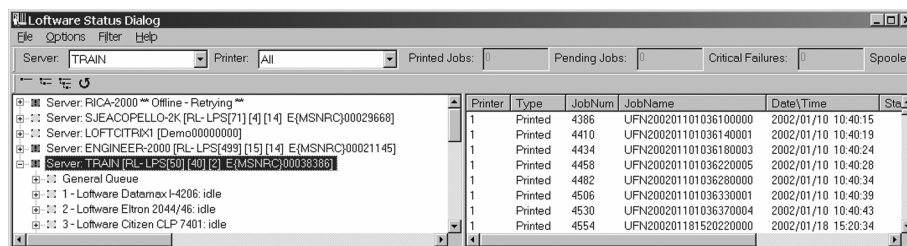


Figure 2-Q: Monitoring multiple servers

Status Example 4 – Interpreting Error Conditions

This example shows a server named RICA-2000 that has five configured printers.

- The icon to the left of the server name is a red exclamation point (!) because there is an error on one of its printers.
- The icon to the left of Shipping Printer 15 is red because it is the printer with the error.
- Note that the error message is given in both the tree and the context views.
- Once the error has been corrected, the job marked as pending for that printer is submitted.
- When one of the icons changes to yellow, it denotes a critical failure. Red denotes that a job did not print because of an error at the printer.

Note: Correct the printer error and print the entire job to change the red icon to green again.

- Note that Printer 15 has an alias defined. This is helpful when reading the tree. (See Chapter 2 in the LLM User's Guide for information on creating alias names for your configured devices.)

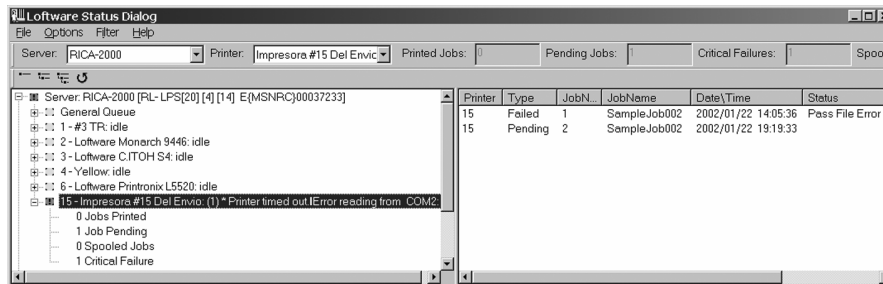


Figure 2-R: Status Application showing a printer with an error condition

Status Menu Choices

File Menu

File | Specify Server – Allows you to add another LPS server to the tree view. You must know the IP address of the server in order to do this. A better way would be to select all servers that you want to view from the server arbitration dialog box explained earlier. You may have to use this dialog box if your LPS server is on a different subnet.

File | Exit – This exits the application.


Options Menu

Options | Expanded – Expands or collapses the tree view on the left to include all server(s), printer(s) and Print Jobs (Printed, Pending and Critical Failures).


Alternatively, click.



Options | Expand Printers – Expands or collapses the tree view on the left to include all server(s) and printer(s).

Alternatively, click. 

Options | Collapsed – Collapses the tree view on the left to include only the server(s).

Alternatively, click. 


Options | Language - This command opens a dialog that allows the user to change the language of the Status Application. As stated in Chapter 1 of the LLM Guide, the selection of the default language for Loftware applications (Design, On-Demand, Range, Batch, etc.) is made during the original installation. However, there may be cases where the default language of Loftware applications needs to change with different users.

Example: The application language is set to “English” during the install. This has worked well for the person doing the installation, “User A”, the next logged in user, “User B”, and the third logged in user, “User C.” However, Users “D”, “E” and “F” would like the language of the LPS Status Client application to be displayed in their native language, French. To change this setting for “User D” and the subsequent users, “E” and “F” in this case, perform the following steps:

1. Press **Options > Language** from the menu bar.
2. Select Français (French) from the drop-down list.
3. Press and hold **Ctrl, Shift and L** simultaneously on the keyboard.
4. Check the box that is displayed – “Set as Default Language for New Users”

This changes the default Status Client application language for Users D, E, and F to French, but it does not affect Users A, B, and C, whose Status Client language remains English. This setting may be changed as often as needed, but each previous logged in user’s settings are retained unless subsequently changed in the **Options > Language** menu.

Options | Refresh - This command refreshes all of the counts for all of the servers and queues. There may be a slight delay if the server is busy when it receives this command. If a server goes offline, (red) and then comes back on-line, clicking the Refresh Icon forces a re-connect immediately as opposed to waiting for the polling refresh. The server displays as “green” right away.

Alternatively, click. 

Note: Refer to ‘Understanding Screen Layout’ above for more detailed information on Polling and screen refreshing.

Filter | Printed – Unchecking this option causes all printed jobs to be suppressed from the context view. This makes it easier to spot pending jobs and jobs with errors.

Filter | Pending - Unchecking this option causes all pending jobs to be suppressed from the context view. This makes it easier to spot printed jobs and jobs with errors.

Filter | Spooled - Unchecking this option causes all spooled jobs to be suppressed from the context view. This makes it easier to spot printed jobs, jobs with errors, and pending jobs.

Filter | Critical Failures - Unchecking this option causes all jobs with errors to be suppressed from the context view. This makes it easier to spot pending jobs and printed jobs.

Note: The more jobs you filter, the faster (and less informative) the tree. Filtering job types that do not interest you (like 'Printed' in many cases) can make a big difference in refresh speed, especially when driving several hundred printers.

Help | Status Specific Help – Brings you to Status Client Application information.

Help | Index – This brings you to the help index of the User's Guide.

Help | About – This brings up an "about" dialog box with the version number.

Server - Drop-down list displays the server(s) to which you are connected.

Printer – Drop-down list displays a specific printer, or all the printers connected to the chosen server(s).

Printed Jobs – Number displayed is the number of jobs that have printed for the selected printer.

Pending Jobs – Number displayed is the number of pending jobs for the selected printer.

Critical Failures – Number displayed reflects the number of Critical Failures for the selected printer(s).

Spooled Jobs - Number displayed reflects the number of Spooled Jobs for the selected printer(s)

Context Menu

Context | View File – Launches the File Editor, which opens the job's file, be it a Pass, CSV, Xml or Command and Batch file. You can then edit the file and resubmit it to the LPS server. See File Editor documentation below.

Context | Resend Job – Resends the selected job to the LPS server. Jobs marked as 'Pending' cannot be resent.

Context | Resend Selected – After performing a multiple selection, resends the selected jobs to the LPS server. Jobs marked as 'pending' cannot be resent.

Note: Refer to the 'Understanding the Screen Layout' section above for more information on deleting jobs.

Context | Delete Job – Deletes the job from the LPS server along with its files.

Context | Delete Selected – After performing a multiple selection, deletes the selected jobs from the LPS server along with their files.

Context | Delete All – Deletes failed and printed jobs for whatever you have selected in the tree view.

*Note: 'Context' is the menu that is displayed after **right clicking** an entry in the context view.*

Using the 'Launch Browser' Feature

If a printer listed in the Status dialog window has been configured in Loftware with an IP address, you can open a Print Server Web Page for the printer in a browser window. To do this, right-click on the printer and press Launch Browser.

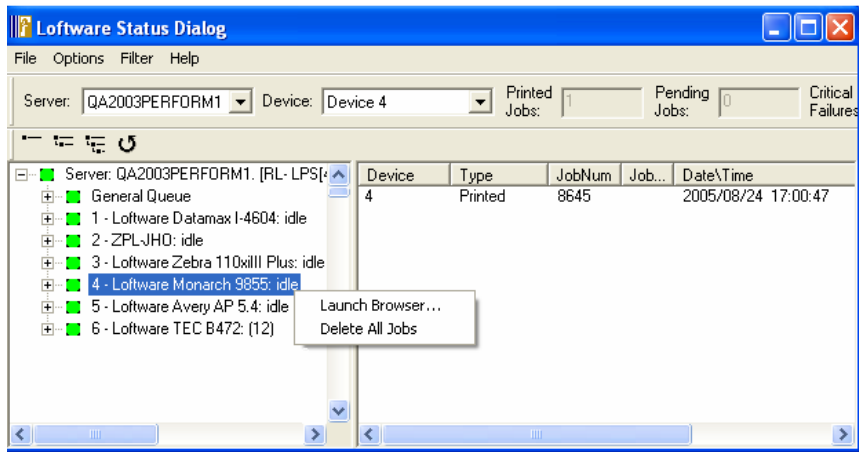


Figure 2-S: Launch Browser Option

The Printer Print Server is launched in a browser.



Figure 2-T: IP Addressed Printer displayed in a Browser

Click on the hypertext links to view information, change configurations, etc., on the printer.

File Editor

The File Editor is launched by right clicking on a printed job on the Status dialog and selecting View File. It is used to make changes to a job's file and resend the job. This is very helpful when your system is in its beginning stages and you want to debug and fix your requests 'on-the-fly' from the status program.

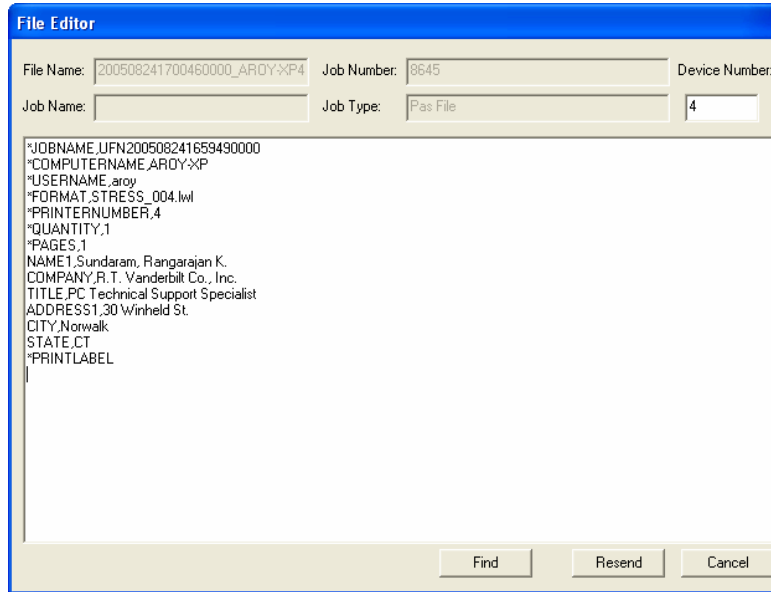


Figure 2-U: File Editor dialog box

Note: To change the printer to which the job was assigned you **MUST** change the Printer Number edit box rather than the file, as the file itself is **NOT** re-processed!

LPSSend Client

What is LPSSend? - A sample program used as a tool to demonstrate the TCP/IP Socket Interface of the Software Print Server. This program opens a socket connection to the LPS, sends a print job, receives status of the job from the LPS and then disconnects. The source code for this utility is included and is available for your use.

For an in-depth discussion on using the TCP/IP interface and viewing the source code for LPSSend.exe, see Chapter 1.

Palm OS SDK

Software has recognized the need for SDKs (Software Development Kits) that operate using the Palm Operating System, and as such, has developed an SDK and sample program for the Palm OS using a Symbol™ hand-held device. The SDK allows the Symbol device to communicate with the

LPS, sending label requests, scan information and print job requests. The sample program described below is intended solely to give the user an idea of how this program works with the Loftware Print Server. After running the sample program, it is suggested that you modify the appropriate files and the code therein and tailor it to suit your individual needs. Again, the program is designed only to give you a small sample of the program and its capabilities.

To view the files associated with this program, in Windows Explorer, go to **Loftware Labeling\Sample Programs\Palm OS SDK\SDK Example Program**. The files listed in this folder are explained below.

LPS_Palm_SDK.H – This header file contains the function declarations for all the functions that you are able to use with the LPS.

LPSPalmSDK.lib – This is a library file that contains all the functions that run the sample program (the SDK).

LPSSend.prc – This is the executable that needs to be downloaded to your Symbol hand-held. You must have a synchronization program in order to do this.

LPSSend.mcp – This contains the project file of our sample code that uses our library. You must have Metrowerks' Code Warrior™ in order to view this file.

Rsc Folder – This is a resource directory containing the resource files for the sample program.

Src Folder – This is a source directory that contains the source files for the sample project (the .c and .h files).

Running the Sample Program

This sample program is designed to let you see how the SDK works. After completing the following steps, there are limitless ways to make the code your own.

1. Install the **.prc** to the Symbol Hand-held device using a synchronization program.
2. Start the LPS on your computer.
3. Click on the LPS Icon on the Symbol hand-held device.

The LPS Configuration screen is displayed:



Figure 2-V: Palm OS SDK Configuration Utility

4. Add your LPS IP Address. *Example:* 172.16.0.128
5. Ensure that **Symbol** is the Labels subdirectory displayed; click OK.

***Note:** The default Labels sub- directory is “Symbol”, but you must be sure that this directory exists on your PC before clicking OK. If the sub-directory line is empty, the default directory is “Labels.” If a new sub-directory is added, it becomes the variable directory for that run of the program.*

At this point, the program enables the Symbol Bar Code Scanner (using the Symbol SDK library functions) connects to an access point on the server (such as the Symbol Spectrum 24™), then connects to the LPS using the specified address, and displays the LPSSend Utility.



Figure 2-W: Palm OS SDK Send Utility

The drop down list in the previous figure is generated using some of the SDK functions, the first of which is the “get label formats” function. The Symbol device goes out to the LPS and asks for a list of label formats in the specified sub-directory, gets that list back, and displays a drop-down list.

6. Choose any one of the three sample label formats from the drop-down list.

This calls another SDK function called “get tab order,” which then communicates with the LPS regarding the format selected, and requests the list of the names of the field(s) on the label. Either the first field name or the key field is displayed on the screen.

7. Enter a key data range from 10000 to 10099.

*Note: The sample labels in the Symbol Directory are all linked to the GM1724.mdb database in the **Loftware Labeling/DBases** directory via the Palm ODBC System Data Source. This file is created when Loftware is installed.*

Printer - A printer number is entered here. Make sure you have a configured printer and have merged the label format to this printer. (See Merging Label Formats in Chapter 2 of the LLM User’s Guide.)

Quantity – Enter a number here, or leave the default at 1.

Duplicates – Enter a number here, or leave the default at 1.

Wait for Status – If this box is checked, the Symbol device sends out information to the LPS, the LPS processes this information, and sends a response back. A response is displayed at the bottom of the Symbol device, such as “Printed.” If the box is unchecked, the LPS responds right away with the job number and processes the request. The Symbol device displays a message similar to: “No Status Available for Job# 22564.”

*Note: **Pros and Cons** - While it is helpful at times to see the status of your jobs to verify that they are being processed, there may be times when this is not practical. Example: If you are out in a warehouse and the Loftware Print Server is very busy, waiting for a status response from the LPS may be very time consuming! In this case, leaving this option unchecked results in sending jobs faster.*

Print on Scan – When this box is unchecked, the Symbol device populates the key field, then prints the label after the user has pressed “Print.” When the box is checked, the label is printed immediately upon scanning.

8. Print some labels using different options selected.
9. Exit the system by going back to the main menu, or by beginning another program on the Symbol hand-held device.

When exiting, the device disables the scanner, logs out of the LPS, and disconnects from the access point.



Chapter 3 Internet Printing

Internet Printing Overview

Loftware recognizes that today's businesses are becoming increasingly geographically diverse and economically streamlined. As a result, there is a need to have robust "thin" applications that allow for data connections that work with ease, that are fast, reliable, and do not require months of up-front work to implement. With Loftware's Internet Printing Applications, a company can maintain centralized control over label printing needs, while sending or receiving print requests from Client-side PCs with attached printers to the LPS via the internet to any of their satellite offices, production warehouses, etc., the world over. Imagine an assembly plant in Taiwan connecting with the home office in Chicago at any time day or night and printing the labels needed for immediate shipment. Loftware has created definitive Internet tools to make this a reality!

Loftware's Internet Printing technology consists of these main areas:

- **Loftware's WebPush / Web Listener** (*i-Push*) technology allows one *server-side* application to control label printers anywhere in the world. The Loftware Web Push "pushes" print streams across the Internet to a Web Listener application. The Web Listener is a client-side application that receives "pushed" print streams from the Loftware Print Server (LPS) across the Internet and prints to locally configured or TCP/IP connected Printers. With *i-Push*, no client site intervention is needed; all the label requests are triggered from a server side application. Labels are "pushed" to vendor sites or satellite offices across the Internet, without the need for expensive WAN connections!
- **Loftware's WebClient** (*i-Pull*) is a thin On-Demand Print client that initiates a request to generate the label. The Web Client connects to the Loftware Print Server (LPS) and prints across the Internet. The difference between Web Push technology and Web Client (*i-Pull*) is that with the Web Client, the print request is triggered from the *client site*.
- The **Internet ActiveX Control** ("*iX*") is a third component of Internet Printing that allows businesses with their own applications to interface with the LPS and print to Client-side printers across the Internet. This utilizes an ActiveX interface across the Internet to print labels to any client-side printer. Information about the ActiveX Internet Control is documented in Chapter 6, but please read the definitions and the Web Server section in this chapter before proceeding to use *iX* and the other two Internet Applications.

Internet Printing Considerations

Performance Considerations

The Performance Considerations section of Chapter 1 (Loftware Print Server) contains a great deal of information about things to consider when developing the type of label and bar code printing solution that gives you the most benefit. Please review that section before continuing with the installation of the WebClient (*i-Pull*) or the Web Listener (*i-Push*). In addition to the information

regarding hardware, number of labels printed, label content and Internet considerations such as that listed below must be taken into consideration:

- Graphics
- TrueType Fonts
- Internet connection speed
- The type of Web Server
- Speed of the client computer
- How fast the client-side printer can image the data
- How many clients and printers are being driven simultaneously

If you have a serious load on the LPS already, adding more load with 500 WebClients or 500 Printers connected to one Web Listener (i-Push) may not be a very workable situation!

Only one LPS can be placed on a single Web Server; however, two Web Servers could each have a LPS, and the load could be balanced between the two. Loftware compresses the data streams to maximize throughput, but we cannot tell you exactly how long labels take to print using the WebClient (*i-Pull*) or using the Web Push/Listener Application. (This is very dependent on a number of variables discussed later in this chapter.) However, we can tell you that our tests here at Loftware have shown that a GM1724A Label, which contains some extremely intricate fields, is retrieved from the LPS, sent back to the client printer over the Internet, and printed in approximately 6 seconds. With the Web Client, this time is further decreased after the first time the label has been downloaded, as it is 'cached' on the client-side computer for future reference! (Read about caching later in this chapter.) With fast hardware and Internet connections, labels can be printed in a number of locations with blazing speed.

Note: Loftware has a web server that you can try out! Send us a test label, or contact Loftware Technical Support or Loftware Sales to see how to do that.

Licensing Considerations

The Internet Printing Applications are licensed similar to the way in which other clients are licensed, that is by seats. The LPS Premier comes with all the Internet Printing modules. More seats may be added on the fly by calling your sales representative or Loftware Customer Service and adding seats. Please remember that once a client computer has logged onto the LPS using one of the clients, that seat is considered taken, and even though that client may be offline, that seat is not available to use by any other client computer. If the LPS is stopped, the licenses are cleared, and a different computer may use a client seat at that time. See the [Loftware Website](#) for more specific information on licensing.

The WebClient

The WebClient (*i-Pull*) application is a **web-enabled** full-featured on-demand printing application allowing users to harness the power of Loftware's Printing Engine (LPS). Detailed, customized operator input screens are created in Loftware's Design Module and presented to the users in the same fashion as On-Demand Print and On-Demand Print Client.

Please note that a browser is not needed to connect to the Internet. Although a small installation must take place, the power and speed gained by using a full C++ application far outweighs the fact that an install must be done. Do NOT be fooled by the limitations of a browser-based solution. Loftware's goal has always been to provide our customers with high quality printing applications, and we believe that goal has been met with the WebClient. The following figure illustrates the concept of the WebClient.

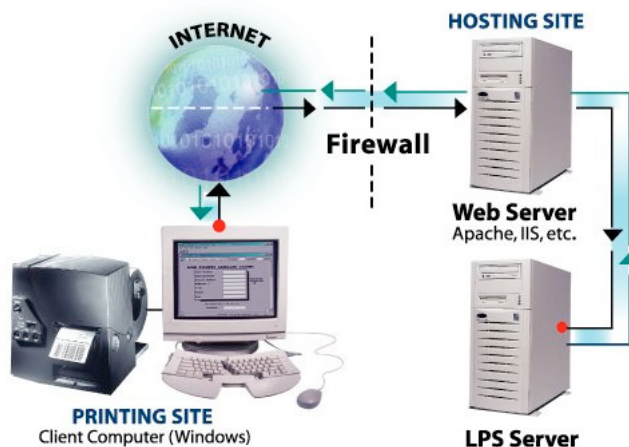


Figure 3-A: The WebClient (i-Pull) Design

As you can see, the Loftware Print Server and the Web Server are shown as separate entities. They are separate to ensure that print speed is not compromised due a heavy load on the Web Server. This is the recommended setup. The firewall shows that communication through the firewall is possible for companies that have this protection on their network. The data going from LPS to the Clients is compressed (sent as a print stream). Also depicted in this figure are a Windows PC, printer, and Internet connection that are all needed on the Client side to print.

How the WebClient (i-Pull) Works

Similar to Loftware's On-Demand Print Client, the WebClient (i-Pull) does not need the Loftware Engine installed on the Workstation PCs that are printing labels. This saves costs, installation time, maintenance and designing/configuring the same labels repeatedly on many PCs. Loftware has developed the WebClient to provide users of the LPS system the ability to select and print labels from anywhere in the world via a low cost Internet connection.

Note: The WebClient (i-Pull) application is licensed by "seats," not "concurrency." This means that the first workstations to log in are recorded as a license taken by the LPS. Once the maximum number of licenses is used, no other WebClient modules are able to execute, even if one logs out. If you want different workstations to be able to use client modules, you must stop the LPS Service and then restart it. This clears the license list kept by the LPS. For complete information on licensing for all Loftware Products, refer to the Loftware Website.

Choosing the WebClient or On-Demand Print Client

When choosing which application to implement, consider the following: The On-Demand Print Client runs on a LAN or WAN, and does not need a web server. On the other hand, the WebClient runs across the Internet, but a web server is required to run it. The following table lists the requirements needed for On-Demand Print (both traditional and client) and the WebClient.

Application	Requires LAN/WAN	Requires LPS	Requires Internet	Stand-Alone	Requires Web Server
On-Demand Print	No	No	No	Yes	No
On-Demand Print Client	Yes	Yes	No	No	No
WebClient	No	Yes	Yes	No	Yes

Definition of Terms

This chapter may contain many terms that are new to you. Before installing or using the WebClient (*i-Pull*), the Web Listener (*i-Push*), or the Internet ActiveX Control (*iX*), please take time to familiarize yourself with these terms.

Apache - A public domain Web server developed by a loosely knit group of programmers. Because of its sophisticated features, excellent performance, and low price (it is free), Apache has become the world's most popular Web server. By some estimates, it is used to host more than 50% of all Web sites in the world. The original version of Apache was written for UNIX, but there are now versions that run under Windows and other platforms.

IIS - Short for *Internet Information Server*, Microsoft's Web server that runs on Windows NT platforms. Because IIS is tightly integrated with the operating system, it is relatively easy to administer. However, currently IIS is available only for the Windows NT platform, whereas Netscape's Web servers run on all major platforms, including Windows NT and UNIX.

Java - A high-level programming language developed by Sun Microsystems. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as *Java Virtual Machines (JVMs)*, exist for most operating systems, including UNIX and Windows. Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web.

Java Virtual Machine - A Java interpreter. The Java Virtual Machine (JVM) is software that converts the Java intermediate language (bytecode) into machine language and executes it. The original JVM came from the JavaSoft division of Sun. Subsequently, other vendors developed their own; for example, the IBM Virtual Machine is IBM's Java interpreter. A JVM is incorporated into a Web browser in order to execute Java applets. A JVM is also installed in a Web server to execute server-side Java programs. A JVM can also be installed in a client machine to run stand-alone Java applications

JDK - (**J**ava **D**evelopment **K**it) A Java software development environment from Sun. It includes the JVM, compiler, debugger and other tools for developing Java applets and applications. Each new version of the JDK adds features and enhancements to the language. When Java programs are developed under the new version, the Java interpreter (Java Virtual Machine) that executes them must also be updated to that same version. You must have **Java Version JDK 1.3** (only) installed

on your Web Server prior to adding the Servlet Engine. [*Not the JRE!*] This may be found at <http://java.sun.com/j2se/>

JSP – Acronym for JavaServer Pages. Created by Sun, this is a way to write snippets of servlet code directly within a static HTML Page. Blocks of servlet code are called *scriptlets* and may use one or more of four variables: request, response, out, and in. The extension for JSPs is .jsp.

Linux - A freely distributable open source implementation of UNIX that runs on a number of hardware platforms, including Intel and Motorola microprocessors. It was developed mainly by Linus Torvalds. Because it is free, and because it runs on many platforms, including PCs, Macintoshes and Amigas, Linux has become extremely popular over the last few years.

Servlet – A servlet is a generic server extension that can be loaded dynamically to expand the functionality of a web server. Servlets are commonly used with web servers, and run within a Java Virtual Machine (JVM). Since servlets are all handled by separate threads within the web server process, they are very efficient and scalable. Servlets are supported on ALL platforms that support Java, and servlets work with all the major web servers. Loftware has developed a servlet called the **LPS Web Servlet** for use with the WebClient (*i-Pull*). Information and installation procedures follow.

Servlet Engine – Servlet Engines are designed to test and deploy servlets. Your choice of servlet engine depends upon the Web Server you are running. Keep in mind, however, that Loftware has developed the LPS Web Servlet to operate ONLY within **Java Version JDK 1.3**.

Unix - A popular multi-user, multitasking operating system developed at Bell Labs in the early 1970s. UNIX was designed to be a small, flexible system used exclusively by programmers, and was one of the first operating systems to be written in a high-level programming language, namely C. Historically, it has been less popular in the personal computer market, but the emergence of a new version called *Linux* is revitalizing UNIX across all platforms.

URI - Short for *Uniform Resource Identifier*, the generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI. (See URL info below)

URL - Abbreviation for *Uniform Resource Locator*, the global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located. For example, the two URLs below point to two different files at the domain *loftware.com*. The first specifies an executable file that should be fetched using the FTP protocol; the second specifies a Web page that should be fetched using the HTTP protocol:

`ftp://www.loftware.com/stuff.exe`

`http://www.loftware.com/index.html`

Web Server - A computer that delivers (*serves up*) web pages. There are many Web server software applications, including public domain software from NCSA© and Apache©, and commercial packages from Microsoft©, Netscape© and others.

Components of the WebClient

The concept of a Web Server and a Servlet is similar to the following:

Java Servlet running on a Web Server

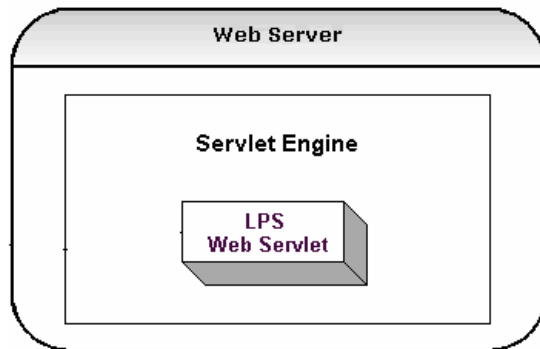


Figure 3-B: Web Server and Servlet Concept

The following components are needed to run the WebClient (*i-Pull*):

WEB SERVER – As stated previously, a Web Server is a computer that delivers (*serves up*) web pages. Every Web server has an IP Address and possibly a domain name. For example, if you enter the URL `http://www.loftware.com/index.cfm` in your browser, this sends a request to the server whose domain name is `loftware.com`. The server then fetches the page named `index.cfm` and sends it to your browser.

- Any computer can be turned into a Web server by installing server software and connecting the machine to the Internet. The Web Server must be up and running prior to any installation of servlets, etc.

Note: *Loftware cannot help you set up or troubleshoot your Web Server!*

- You may set up restricted access (security) to your Web Server, which requires a User Name and Password when starting Loftware’s Web Client. The Web Server secures or protects a resource, like the Loftware URI.
- Think of your Web Server as the ultimate “Ringmaster”, directing the action of hundreds of “client rings” wherever they may be, in the next room, or on the next continent. Some common Web Servers and Servlet Engines are listed below:

Apache Server	httpd.apache.org
Microsoft IIS Server	microsoft.com/ntserver/techresources/webserv/default.asp
Netscape Enterprise Server	wp.netscape.com/enterprise/v3.6/
Sun's JDK	java.sun.com/j2se
Allaire's JRun	macromedia.com/software/jrun
Jakarta's Tomcat	jakarta.apache.org/tomcat
A site with all major Web Servers listed, complete with Operating System (OS) information, prices and versions.	redir.internet.com/cgi-bin/quickcompare.pl

SERVLET ENGINE – Servlet Engines take one of three forms:

- **Stand-alone** – A standalone engine is a server that includes built-in support for servlets. These work well for initial tasking, but usually lack the power of a dedicated web server.
- **Embeddable** – An embeddable engine is a lightweight servlet deployment platform that can be embedded in another application.
- **Add-on** – An add-on servlet engine functions as a plug-in to an existing server. It adds servlet support to a server that was not originally designed with servlets in mind. For many companies (including Loftware) that already have servers, this type of servlet engine is often a good choice. The add-on servlet engines utilized by Loftware in the development of the WebClient are Jakarta's "Tomcat", found at <http://jakarta.apache.org/tomcat/> and Allaire's "JRun 3.0", found at <http://www.allaire.com/Products/JRun/>

Loftware does not recommend or endorse any particular servlet engine, just as we do not endorse any printers. There are many servlet engines to choose from, and your selection should be based on what engine serves your company's needs best. It is similar to buying a car, in that thought must be taken as to the features that are needed most. Take your servlet engine for a "test drive" first to see if it works well for your application.

*Note: When setting up the JSP/Servlet container, make sure the Java VM path is set to **Java Version JDK 1.3** (currently Java 2 SDK) or higher. **Do not use the JRE; use the JDK/SDK download.** This can be downloaded from <http://java.sun.com/j2se/>.*

Tomcat IIS How To – If you intend to use IIS with Tomcat, please go to this link:

<http://jakarta.apache.org/tomcat/tomcat-3.2-doc/tomcat-iis-howto.html>

In the section entitled "**Adding additional Contexts**", the new context settings should read:

```
/loftware/*=ajp12
/loftware=ajp12
```

Follow the rest of the instructions in the document for some very helpful information on using IIS with Tomcat.

LPS WEB SERVLET - The LPS Web Servlet is found in Loftware Versions 5.5 and above, in the LPS Premier Edition. The LPS Web Servlet is Java-based web application that enables data communications from the WebClient to the Loftware Print Server through the Internet. The LPS Web Servlet Application has two main functions:

- The first main function consists of a Java Servlet that facilitates the communications between the WebClient (i-Pull) and the LPS. Why did Loftware choose a Java Servlet to accomplish this? Servlets are powerful, portable, efficient, and have great endurance. Since it is written in Java, the LPS Web Servlet can go from being deployed on a Windows NT machine to a high-end Unix server. URL access, multi-threading, data compression, and database connectivity are all part of this package. Using the Internet as the conduit, when the WebClient is opened, the URL configuration (address) lets the WebClient know what server to go to. The WebClient establishes a virtual connection with the Loftware print server using the LPS Web Servlet as a mediator. Transactions are marshaled by the WebClient throughout the [Web] Client / Server [LPS] interaction. The LPS gathers the requested data and sends it back to the LPS Web Servlet, and the Web Servlet then sends the data back to the WebClient. This description may sound like a lengthy process, but rest assured that transmissions take place as fast as your Internet connection allows!
- The second main function is administrative. It is displayed through a JSP page used in a browser such as Internet Explorer or Netscape and is an interface to the **LPS Web Servlet** for troubleshooting and diagnostic purposes. The page contains critical configuration data about the servlet and the LPS. The LPS Web Servlet has been packaged as a *Web Archive file* (.WAR, in this case, **loftware.war**)

A Web Archive file is a zip (jar) file that takes the form of a standard file archive format for the Java Platform for deployment to a Java-enabled web server. War files are compatible with all JSP containers that comply with Version 1.1 of the JSP specification. The examples used in this chapter are Jakarta's Tomcat 3.1 and Allaire's JRun 3.0.

Note: The LPS Web Servlet application does not have to run in a Windows environment; it can run on Linux or a Unix box without Windows.

LOFTWARE PRINT SERVER - The LPS is described in Chapter 1. The LPS forms the basis of Loftware's "Server-Centric" approach to bar code labeling systems. Server-Centric means that all bar code label printing in an area, building or enterprise is controlled from a centralized PC on the network.

- The LPS is capable of printing labels from your ERP/MRP II and/or WMS systems, regardless of the platform on which they reside.
- Host computers and operating systems, such as UNIX, AS/400, HP 9000, DEC VAX and Risc/6000 can also request labels through the LPS. It is also a viable solution for client/server PC and RF applications.
- The WebClient (*i-Pull*) is used in conjunction with the LPS, gathering label information, such as data, layout, design, etc. It is suggested that you install the LPS to a different machine than the one on which the Web Server resides, so if either of these applications develops a problem, the other component is not affected. (See the Performance Considerations section on the Loftware Website or in Chapter 1 of this guide for more information on this)

Installing the WebClient Components

If you have had previous experience installing Web Servers and/or Web Servlets, the following steps may seem fairly straightforward and easy to accomplish. However, if you are new to all of this, please take some time to read this chapter and the preceding two chapters carefully, as well as

taking time to read the information on the Websites that have been mentioned in this chapter to give you a clear idea of the complexity of this task. We would like to emphasize that a little up-front research and study may save a lot of back tracking and troubleshooting in the future.

As you continue reading this chapter, complete each step, checking them off as you go:

- **Step 1:** Install LPS (Loftware Print Server)
- **Step 2:** Configure CLIENT DEFINED Printers
- **Step 3:** Install Web Server
- **Step 4:** Install Servlet Engine
- **Step 5:** Install LPS Web Servlet
- **Step 6:** Install WebClient
- **Step 7:** Test Connection to WebClient

Step 1 - Install the Loftware Print Server

If you have not already done so, please install the Loftware Print Server (LPS) as outlined in Chapter 1 of this guide.

Step 2 - Configure CLIENT DEFINED Printers

CLIENT DEFINED printers are printers configured for future use by a client PC. They must be configured on the server, with pre-set Printer-Specific Options, (PSOs) and Label-Specific Options (LSOs). Read Chapter 2 in the LLM User's Guide for more information on CLIENT DEFINED printers.

Step 3 - Install a Web Server

If you do not already have a Web Server, please install one, and get it up and running. Loftware Technical Support cannot help you with the installation of your Web Server, nor help you with any problems related to your Web Server.

Step 4 - Install the Servlet Engine

This chapter contains information about the types of Servlet Engines, their drawbacks and benefits. If you have not already done so, please install a Servlet Engine at this time.

Note: Loftware cannot help you set up or troubleshoot your Servlet Engine!

Step 5 - Install the LPS Web Servlet into the Servlet Engine

It has been stated previously that you may use any Servlet Engine you would like. In this section, Loftware includes examples of installation of the LPS Web Servlet on two Servlet Engines, "Tomcat" and "JRun 3.0". It is very important that you understand your Operating System before proceeding with the installation of the LPS Web Servlet. In addition, please read Tomcat and JRun documentation, especially the troubleshooting sections.

Windows Installation of LPS Web Servlet into Tomcat

1. Copy the **loftware.war** file to the directory where you installed Tomcat, i.e., C:\Jakarta-tomcat\webapps directory.

(**loftware.war** is found on the Loftware CD, under the *LPS Servlets* folder)

2. Start Tomcat using the **startup.bat** file.
After Tomcat comes up fully, shut it down using the **shutdown.bat** file.

During the previous step, Tomcat should have automatically extracted the **loftware.war** file into the War file structure under the Webapps directory. (See Figure 3-D.)

3. Modify the **web.xml** file as outlined in the next section. (This sets the address of the LPS Server.)
4. Restart Tomcat for the changes to take effect.
5. Verify the Web Application is properly installed by loading up a browser and go to **http:// <ipaddress>:port** (if needed –default 8080)/**loftware/ LPSRPT.jsp** and verifying the LPS Web Servlet page is displayed. (See the following Figure.)

This completes the Windows installation of Tomcat and the LPS Web Servlet application. Tomcat should now be configured to a Web Server such as IIS or Apache using the appropriate adapter or connector. Refer to your Web Server's and your Servlet Engine's (such as Tomcat) documentation.

To reinstall the Web Application, go to the **C:\Jakarta-Tomcat\webapps** directory, delete both the **loftware.war** file and the **Loftware** directory and re-install as outlined previously.

Unix Installation of LPS Web Servlet into Tomcat

1. Copy the **loftware.war** file to the directory where you installed Tomcat, i.e., /usr/lib/jakarta-tomcat-3.2.1.
2. Start Tomcat using the **start.sh** file.
After Tomcat comes up fully, shut it down using the **shutdown.sh** file.

During the previous step, Tomcat should have automatically extracted the **loftware.war** file into the War file structure under the Webapps directory. (See Figure 3-C.)

3. Modify the **web.xml** file as outlined in the next section.
4. Restart Tomcat for the changes to take effect.
5. Verify the Web Application is properly installed by opening up a browser such as Netscape or Internet Explorer and go to **http:// <ipaddress>:port** (if needed –default 8080)/**loftware/ LPSRPT.jsp** and verifying the LPS Web Servlet page is displayed.

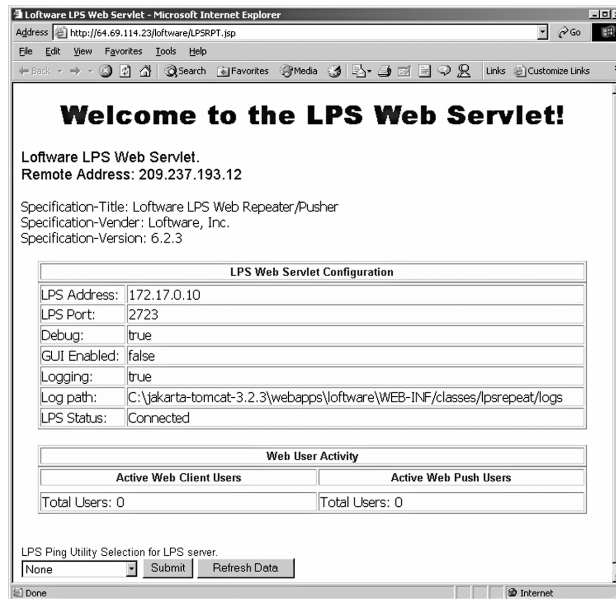


Figure 3-C: LPS Web Servlet Page

This completes the Unix installation of Tomcat and the LPS Web Servlet application. Tomcat should now be configured to a Web Server such as IIS or Apache using the appropriate adapter or connector. Refer to the documentation included with your Web Server and your Servlet Engine. (I.e., Tomcat).

To re-install the Web Application, go to the **C:\Jakarta-Tomcat\webapps** directory, delete both the **software.war** file and the **Loftware** directory, and re-install as outlined previously.

Windows Installation of LPS Web Servlet into JRun

1. Open Jrun on your Server.
2. Click on the JRun default server in the left pane of JRun's JMC.
3. Click on War deployment in the main (right side) pane
4. Enter the following properties in the right pane:
 - a. Servlet War File or Directory: **Path to loftware.war** (i.e., C:\Program Files\Loftware Labeling\loftware.war. The loftware.war file is also found on the Loftware CD.
 - b. JRun Server Name: Select Installation Server (i.e., **Jrun Default Server**)
 - c. Application Name: **loftware**
 - d. Application Host: **All Hosts**
 - e. Application URL: **/loftware**
 - f. Application Deploy Directory: **C:\JRun\servers\default\loftware**

5. Click on **Deploy**.

If unsuccessful, fix any errors and try again.

6. Modify the web.xml file as outlined in the next section.
(This sets the address of the LPS Server)
7. Restart the JRun server.

Verify the Web Application was properly installed by loading up a browser and go to **http://ipaddress:port(if needed –default 8100)/loftware/ LPSRPT.jsp** and verifying the LPS Web Servlet page is displayed.

This completes the installation of LPS Web Servlet Web Application on JRun. JRun should now be configured to a Web Server such as IIS or Apache using the appropriate adapter or connector. Refer to documentation for your Web Server and your Servlet Engine (such as JRun).

Note: The WebClient (i-Pull) only works if you have the LPS configured. See Chapter 1 for information on installing and configuring the LPS.

Note: Be sure to save the changes and restart the JSP container, (i.e., Tomcat or JRun).

To re-install the LPS Web Servlet Application, select the **JRun Default Server/Web Applications/LPS Web Servlet** in the left pane of the JRun Management Console. In the main pane, click on **Delete** to remove the LPS Web Servlet Application, then re-install the Web Application as outlined above.

*Note: After installing and upon re-starting the servlet engine, a directory structure is created. The web.xml file that is created upon the re-start **must** be modified. See following instructions:*

Web.xml file modification

The web.xml file is found in the newly created directory structure, which is seen in the figure below. (Your files may look slightly different, depending on your OS and which servlet engine you have installed.)

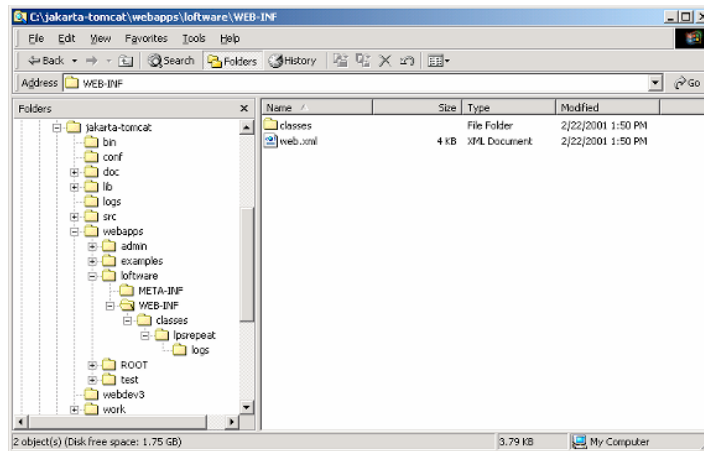


Figure 3-D: Typical Directory Structure of servlet engine after installing in Windows.

After installation of the Loftware Web Application, the servlet configuration file (web.xml) needs to be modified so that the application is able to locate the LPS Server.

In order for the changes to take effect, the servlet engine needs to be restarted.

The WAR file: loftware.war has a descriptor file named **web.xml** that is located in the WEB-INF directory. This file is an XML file that is used to configure the Loftware LPS Servlet Application. The web.xml has a section entitled LOFTWARE Servlet Configuration, in which only one area needs to be configured, that being the IP Address to the LPS.

Configuration Areas

LPS IP Address

Property

Example: 172.17.0.137

The web.xml file looks like:

```
<!-- LOFTWARE Servlet Configuration -->
<context-param>
  <param-name>LPSAddress</param-name>
  <param-value>180.10.0.231</param-value>
  <description>
    Ip Address of the LPS Server.
  </description>
</context-param>
```

▲ (Change only the values in between the > and < tags).

Step 6: Install the WebClient (*i-Pull*)

Install the LPS WebClient to the Client PC one of the 3 following ways:

1. A "Full" install from the CD adds the WebClient executable.
2. An install from the **LPS Clients** folder or **LPS WebClient** folder on the CD.
3. A WebClient install hosted on the Loftware Print Server and downloaded by the Client PC.

Step 7: Test the Connection to the WebClient (*i-Pull*)

1. Design the necessary label formats (using Label Design Mode), making sure that you customize your operator input screen.
2. Configure the printers for these labels as “CLIENT DEFINED” in Label Design Mode and put them on the server. (See the following figure.)

Chapter 1 in this guide and Chapters 2-5 in the LLM User’s Guide have extensive information on Device Connections and configurations.

Important Note: *All configured printers to be used with the WebClient (i-Pull) must be configured as “CLIENT DEFINED,” meaning that they are manipulated by the WebClient and not by the Loftware Print Server. See Chapter 2 in the LLM User’s Guide for information on CLIENT DEFINED Printer configuration.*

3. Launch the Client from a shortcut on the desktop or Start/Program Files/Loftware Labeling/Print Server/Internet Clients/Web Printing Client.

The first time you open the WebClient on the Client PC, the following dialog box is displayed:

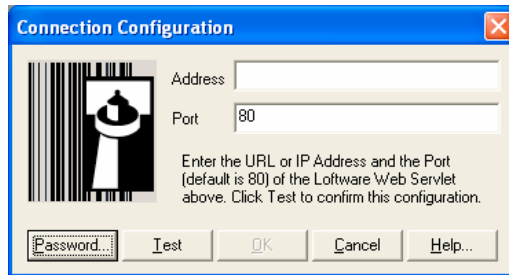


Figure 3-E: Configuring Web Server Connection

4. Enter the IP Address and Port or URL and Port for the Web Server to which you are connecting.

Example: The IP address may be “159.64.0.149”, while a URL may be “www.yourcompany.com”.

If you are unsure of how to proceed, press the help button, and a Help Window is displayed:

5. Press the *Password* button if the administrator has set up security on your Web Server, and enter the authentication information.

Note: *If a password is required, and Test is pressed first rather than Password, a message box is displayed prompting you for your user name and password, and these must be entered.*

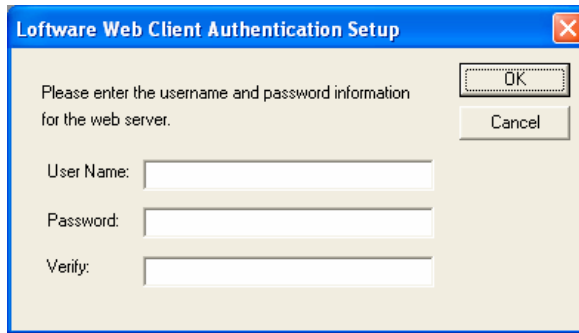


Figure 3-F: Web Client Authentication

6. Enter the Authentication Information, press *OK*, then press *Test*.

Note: The test function is required to ensure that the correct Address and Port have been entered. The **OK** button is **not** enabled until this function has been performed successfully.

7. If the test is not successful, check the error message from the Web Server (example shown in figure below), press *OK*, correct the error, and try again.

Remember: Loftware cannot help you with Internet connections and address problems.

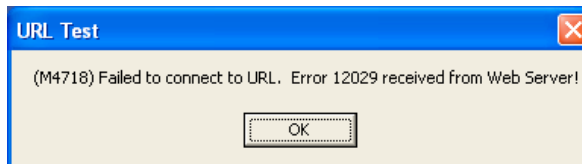


Figure 3-G: Connection Error from Web Server

8. If the test is successful, a “Successfully Connected” message is displayed, press *OK* to close the box.

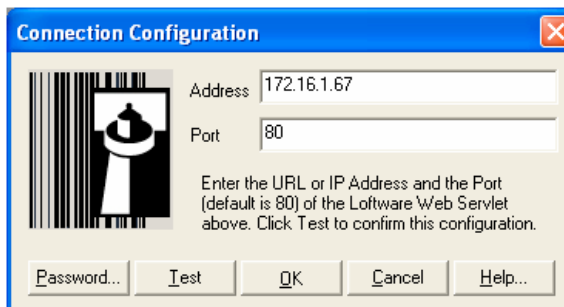


Figure 3-H: Connection Configuration

9. Press *OK* again to open the **File Open** dialog box on the server.

The first time it is opened, the **Loftware WebClient File Open** dialog box is displayed within the WebClient screen. This dialog box contains the list of labels and/or folders that are on the server. This list of labels and/or folders is the *same* as the root directory of the Labels Path on the LPS.

After the first time WebClient is run, the last label opened is displayed, as the label design and device configuration are stored in a cache file once they are opened.

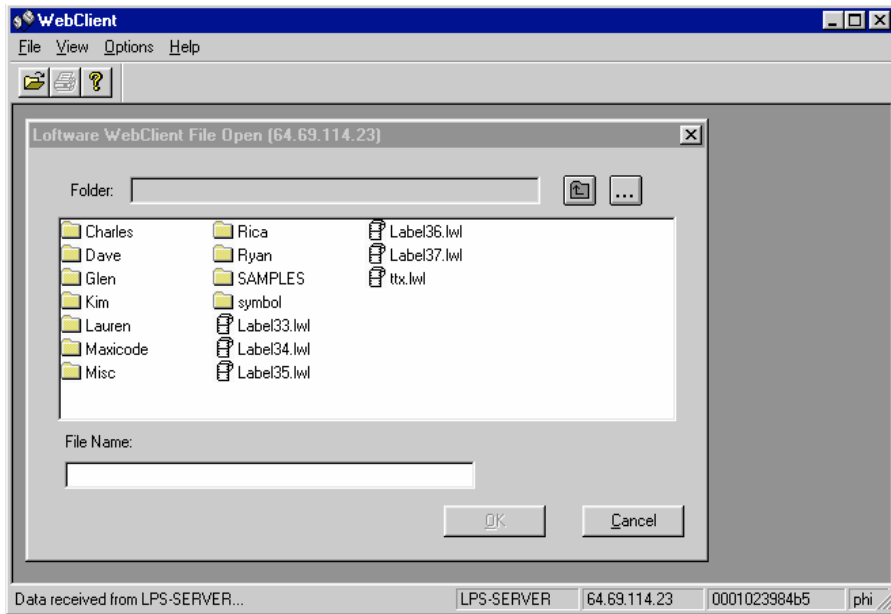


Figure 3-I: The WebClient (i-Pull) window, displaying designed/configured labels

10. Choose the Label you want to print from the Labels Directory.

The first time you open this label, one of the following warning messages is displayed:



Figure 3-J: Request for CLIENT DEFINED Printer for Label

This message above indicates that the printer cannot be locally configured, because it has not been added to the LPS as CLIENT DEFINED. You *must* configure this printer on the LPS before continuing.

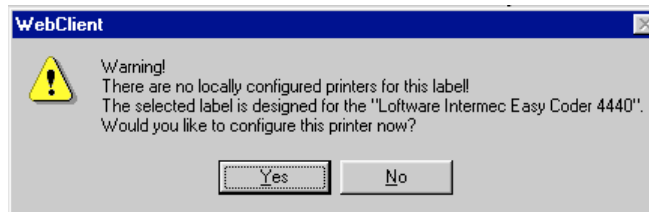


Figure 3-K: Request to locally configure the label.

The second message indicates that the printer does exist on the LPS as CLIENT DEFINED, but now must be configured locally.

11. Click on *Yes*, the Configure Printer dialog box is displayed.

The drop-down list displays all the CLIENT DEFINED configured printers on the Server. (None of the printers have been locally configured.)

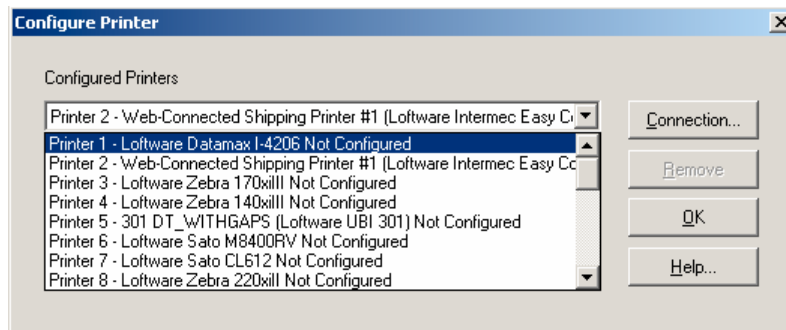


Figure 3-L: Configured Printers List

12. Choose the printer the label is designed for, click on *Connection*.

The following dialog box is displayed:

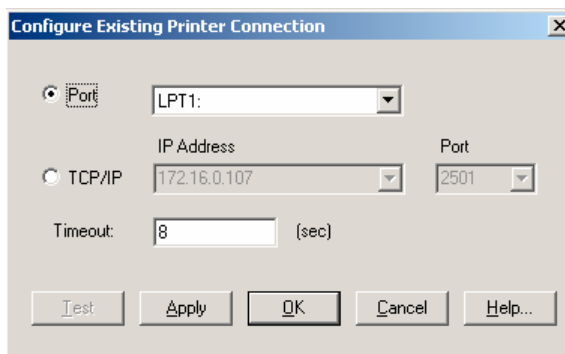


Figure 3-M: Configuring Local Printer Connections

13. Choose the type of connection, Port, or TCP/IP.
(See Chapter 2 in the LLM User's Guide for further information on Connection types.)
 - a. Click on *Apply*, then *Test* to send a test print stream to the printer, or
 - b. Click on *OK*, and *OK* again to close the window and return to the label to begin printing.

Testing your connection is always a good idea the first time you are printing the label!

To change or remove local printer configurations, press *F6* or *File / Devices*.

Note: Remember that in order for a USB Port to be displayed in the Port list, the printer must be connected to the PC and powered on.

Using the WebClient

The WebClient screen is very similar to the On-Demand Print Client screen, as shown below. Note the information displayed in the status bar.

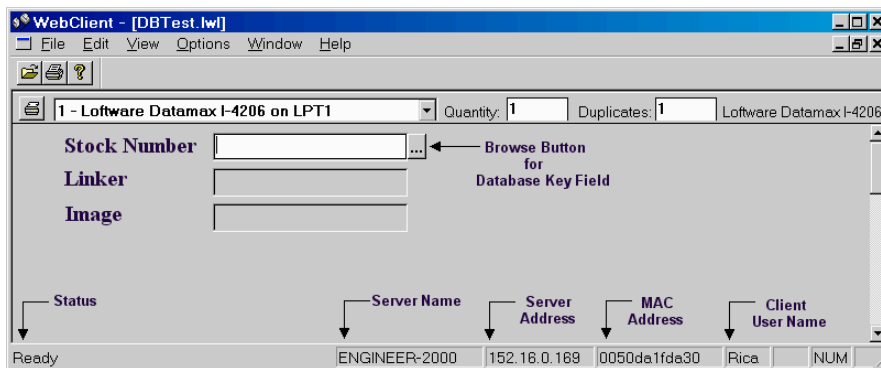




Figure 3-N: The WebClient (i-Pull) Screen

Make sure printers are connected and have been tested prior to this operation!

1. Launch the Client from a shortcut or the 'Start' button as described previously.
The Software WebClient window is displayed.
2. Select File | Open.
The Open dialog box displays the Label File.
3. Double-click the label you want to print from the Label Directory, or select it and click *OK*.
4. Enter information in the data fields manually, or if the label is connected to a database, use the database browse button. (See next section for information on WebClient and databases.)
5. Change the Quantity and Duplicates if needed. (Default = 1)
6. Press the Printer Icon  to print, or *File / Print / OK*.

Using the WebClient (*i-Pull*) with a Database

The WebClient is able to print Serial Numbers, Incrementing/Decrementing Fields, Check digits, Formulas, etc. For complete information on Data Sources, see Chapter 7 in the LLM User's Guide, either the print or on-line version. *Example:* If your label has a Database Key Field, use one of the following options to access the database information for your label:

1. Click on the *Browse* button;  the first 25 records in the database are displayed.
2. Refine the search by typing one or more characters in the Search Key box, click the *Go* button.

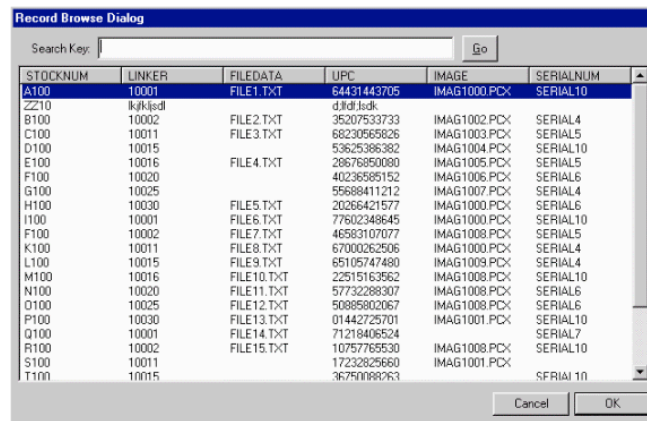


Figure 3-O: Record Browse Dialog Box

- Alternatively, to filter the database, enter one or more valid character(s) in the key field and click on the *Browse* button.

A dialog box with option buttons is displayed:

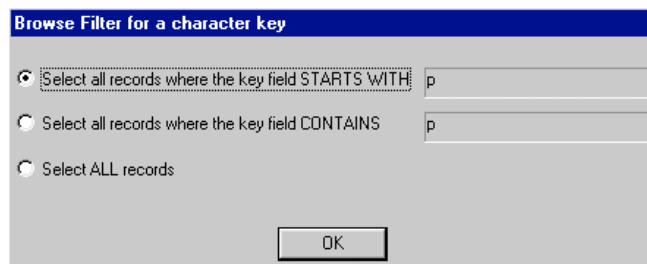


Figure 3-P: Browse Filter

3. Choose one of the three options; the database is “filtered” to display only database fields that start with the character(s), contains the character(s), or all records.

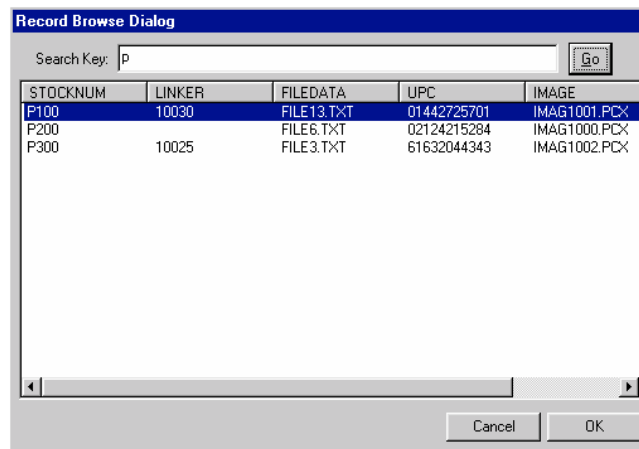


Figure 3-Q: Browsing the database using the Search Key

4. Double-click the record you wish to open.
5. Print the label(s) by clicking on the Printer Icon, select *File / Print / OK* from the menu, or choose *F9 / OK* on the keyboard.

Additional WebClient Information

The Cache Directory

A cache directory is created when the first label downloaded from the LPS is opened on the client machine. Labels that are subsequently opened are also saved in this directory. This keeps you from having to download the label formats, layouts and printer information each time this label is opened. This helps to make the WebClient run with blazing speed after the label formats, etc. have first been received from the LPS. The cache directory is created under **C:\Program Files\Loftware Labeling**, and as you open each label, the directory structure that is created is replicated in the same manner as the label directory of the LPS to which you are connecting.

The Cache Directory may be purged by choosing **Options|Purge Cache Directory** from the Menu bar. You may wish to purge the Cache Directory if you have opened up many labels over a period of time, and they are taking up a lot of disk space on the Client PC. This may be especially useful if you are running the WebClient on a PC with a limited amount of disk space, or if you have opened up a number of labels you are no longer using.

Troubleshooting the WebClient (i-Pull)

Diagnostics

Open the Diagnostics Window by clicking **View|Diagnostics** from the Menu Bar of the WebClient window.

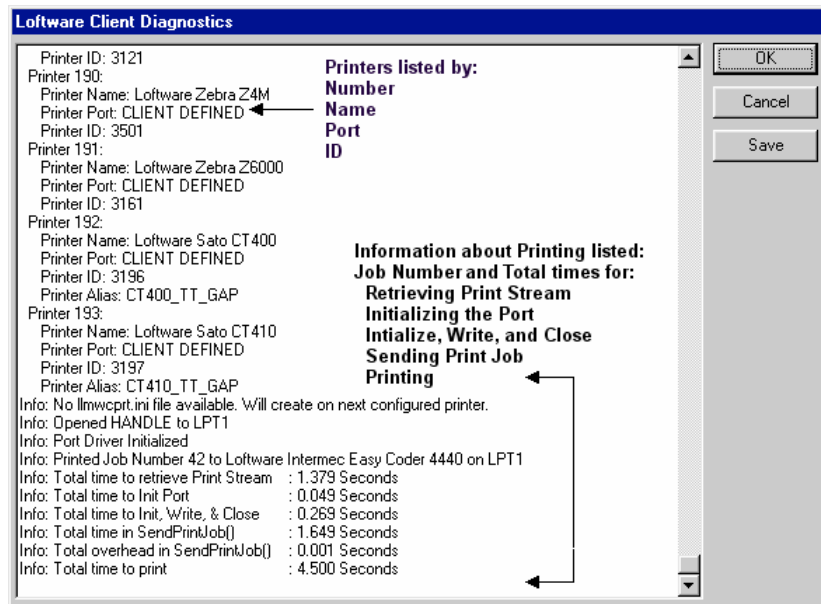


Figure 3-R: Loftware Client Diagnostics

The Diagnostics Window is useful to view to see any errors that may have occurred with printing, server connections, etc. It displays the total printing time as well, which may be quite useful if you are comparing printing time with a dialup connection versus a cable modem connection. Technical Support may view this file to help you troubleshoot printing and/or connection problems as well. If you are still having trouble printing using the Thin Client On-Demand Printing module, read the following checklist before calling Technical Support

1. Has your connection been tested?
2. Verify that the right printer has been correctly configured for your labels. (Both locally and on the LPS.) If it has not, configure the printer.
3. Verify that all server information is correct, such as the IP address, etc.

Make sure that you can test print a label to the selected printer.

Troubleshooting Servlet Engines Tomcat and/or JRun

It is recommended to install Tomcat in the root directory. (On Windows in the C:\Jakarta-tomcat directory and on Unix in the /usr/lib/ Jakarta-tomcat directory.) This makes the paths short and easier to find. Tomcat has excellent documentation that can be found in **jakarta-tomcat\doc\appdev\index.html**. It is recommended to read these documents.

- Did you create an environment variable called **JAVA_HOME** and set it to the JDK 1.3 location? *Example:* JAVA_HOME=C:\jdk1.3
- Did you add **\$JAVA_HOME/bin** to your Path so the Java command is recognized from the command line? This must point to the JDK1.3\bin directory.

- Did you create an environment variable called **TOMCAT_HOME** and set it to the Jakarta-tomcat directory? *Example:* TOMCAT_HOME=C:\Jakarta-tomcat.
- Did you verify that Tomcat is installed correctly by going to a browser and accessing the default **Tomcat JSP page**? *Example:* <http://Address: Port (if needed –default 8080)/examples/jsp>. Click on an execute option and make sure the Tomcat example JSP executes. This tests the installation of Tomcat. If this fails, make sure the above steps are correct.
- Did you install JRun in the root directory? (On Windows in the C:\JRun directory and on Unix in the /usr/lib/JRun directory.) This makes the paths short and easier to find. Also, make sure that JRun points to the **JDK1.3\bin directory**. This value is set during the JRun installation.

More Information - JRun has excellent documentation that can be found by going to: Start Menu/JRun 3.0/JRun documentation. Specific information can also be found under **Deploying J2EE Applications Using the JMC** and the **JRun Quick Product Tour** located in the JRun Management console. Click on **App. Deployment** and then **Deploying a Web Application**. Instructions for installing Web Applications are found here. It is recommended to read these documents.

Troubleshooting Printing Problems with WebClient (i-Pull)

While Printer Configurations are relatively simple when using the WebClient, there are some points to keep in mind to ensure smooth operation of all print jobs. Listed below are some potential problems that may occur when first using the WebClient.

The Printer to which you want to print is not showing up in the WebClient Window – Make sure you have selected the printer for which the label was designed, or the drop-down printer list does not display this printer. Remember that this application prints only the label(s) that has been designed for a specific “CLIENT DEFINED” printer that is configured on the server. You are not able to print the label using another printer when using WebClient on a Workstation PC. If you want the same label to be printed by the WebClient on two different printers, then that label must be created in Loftware’s Design Mode for each printer; both printers must be set as CLIENT DEFINED, and must be available on the server.

Labels are not printing, even though the drop-down printer list is populated – Check your printer connections, review Chapter 2 and Appendix B in the LLM User’s Guide for further instructions on printer errors. For example, check to make sure the right stock has been chosen for the CLIENT DEFINED printer.

Internet Data Push and the Web Listener

Overview of Data Push

Data push is the concept of requesting data from one central spot and, in the case of label printing, having it print anywhere on your LAN/WAN or Internet. Loftware has been performing data push on company LANs/WANs for many years, but *Internet* data push is new. Using advanced technology, client sites can set up printers and receive print requests from the LPS server across the Internet.

Data Push allows one server-side application to control label printers anywhere in the world. You can “push” labels to your vendor sites or satellite offices without the need for expensive WAN connections. The difference between data push technology and our web client technology is that with the web client, the print request is triggered from the client site. With data push, no client site intervention is needed; all the label requests come from a server-side application.

Loftware has termed the Internet data push application Web Push. The concepts of Web Push and the accompanying Web Listener (i-Push) are visually displayed in the following figure:

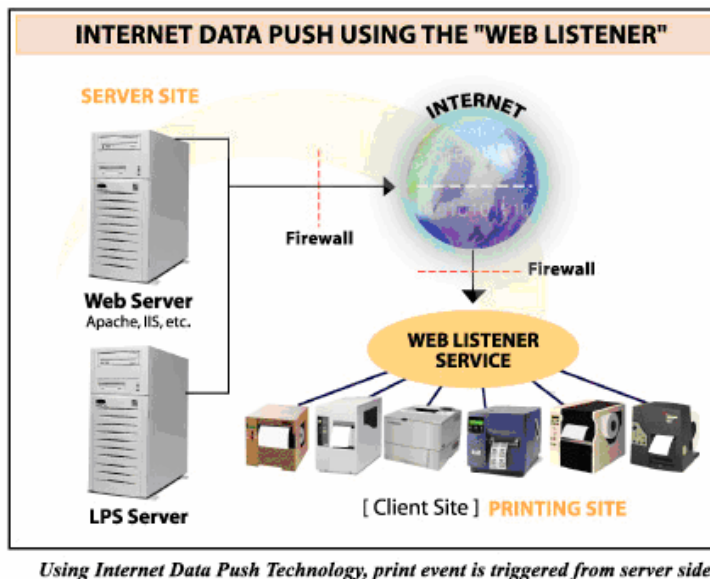


Figure 3-S: Concept of Internet Data Push

Note: One Listener can service as many printers as you would like (or have seats for), but to improve throughput, you may want to divide the load, such as 1 Listener for every 10 printers.

There are three major subparts to the Push system:

The Software Print Server (LPS)

With Web Push technology, the Software Print Server does its usual job of receiving and handling print jobs, but in addition, when a job request is received via a .pas, .csv, .xml file, etc., the LPS forwards notification of the job to the LPSPushervlet. In order for the job request from the LPS to be successful, Internet and printer configuration are required. More information on configuration follows.

LPS Push servlet

LPS Push servlet is a Java Servlet running on the web server that manages the job notifications coming from the LPS. It is a “middle man” so to speak. When the LPS gets a job request, it sends notification to the LPSPushervlet, and the servlet in turn sends the notification of the job to the Web Listener. The Web Listener in turn gets the data stream from the LPS via the LPS Push servlet and sends an update back (again via the servlet) when the print job is complete.

Web Listener

This is the Client-side application that connects to the LPSPushServlet in order to receive Print Jobs. Thus, the name “listener” is apt, because this application “listens” for print streams coming from the LPS via the LPSPushServlet. Initial printer configuration is required; more information on this follows.

Installing the Web Push Components

Installing Web Push / Web Listener (i-Push)

Note: Web Push and Web Listener are only available in the Software Premier Edition.

Step 1: Install the Software Print Server – If you have not already done so, install the LPS as outlined in Chapter 1.

Step 2: Install the Web Server – If you have not already done so, follow the instructions earlier in this chapter regarding Web Server installation. Software cannot help you with the installation of your Web Server.

Step 3: Install the Software Web Servlet – If you have not already done so, follow the instructions earlier in this chapter to install the Software Web Servlet.

Step 4: Install the Web Listener Client - The **Web Listener** is similar to the WebClient in that it may be installed one of three ways:

1. A “Full” install from the CD adds the Web Listener Client executable when you have chosen to initialize the LPS during install.
2. A separate Client install from the Clients Folder on the CD, found under **LPS Internet Installations > InternetPrintingClients.exe**.
3. An individual Web Listener Client install, either from the CD, or one that is hosted on a Web Server and downloaded to the Client PC, called **WebListen.exe**.

You are now ready to configure users and printers, and start Internet printing!

Configuring the Web Push Components

Server and Client-Side Configuration - The Printer Configuration information that follows is a two-part process, whereby the Printers that the Client intends to use are configured on the Server-side first, then the Local or LAN/WAN Ports are configured on the Listener (Client) side.

Configuring Web Push (Server Side)

The Software Print Server keeps a list of all the Listener Clients that can connect. When a Listener connects, the printer configurations on the Server for that Listener are downloaded to the Listener. Each of these printers *must* then be configured with a Port on the Listener Side.

Step 1: Add a Web Push Client to the Server (LPS) database

1. Open Software Design on the PC running the LPS (Server-side)
[**Start > Programs > Software Labeling > Design32**]
2. Open the Configure Printers dialog box
[**File > Devices** or **F6**]
3. Highlight a printer, click on *Connection*.
4. Choose “Web User” in the Print Using section of the configuration dialog box, and make sure that the Shared Network Printing box is checked.

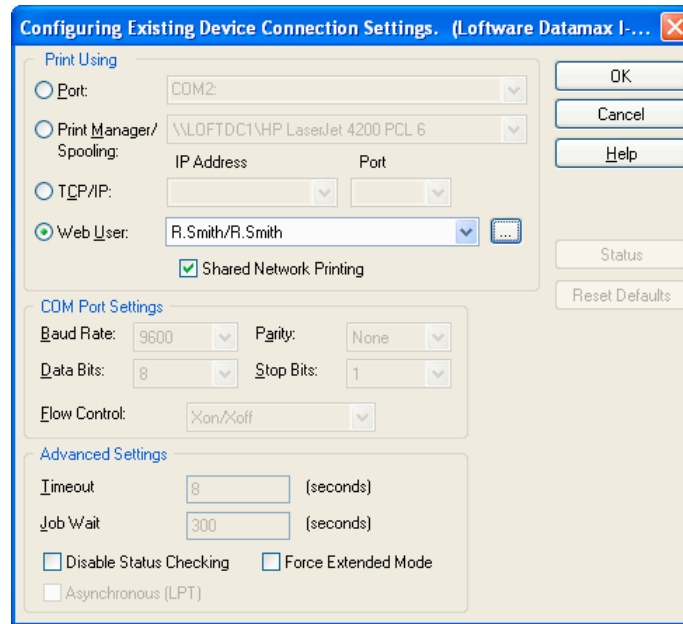



Figure 3-T: Web Push User Configuration

5. Click on the drop-down list and choose the designated user. If the user does not exist on this list, click on the ellipsis button 

The LPS Users dialog box is displayed:

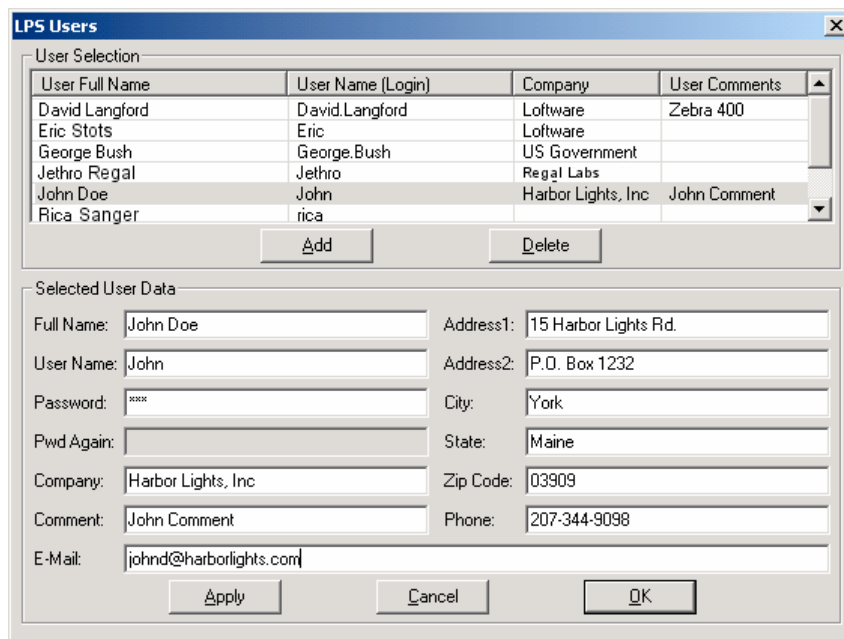


Figure 3-U: LPS Users Dialog Box

6. Click *Add*; fill in the blanks in the Selected User Data section. The User Name is the only required field.

Note: It is suggested to use a full name to avoid confusion. The User Name must be unique, meaning that no two people on the User List may have the same User Name, or a warning message is displayed.

7. Click *Apply* to save the addition and create more LPS Users as necessary
8. Press the *Add* button to add as many LPS users as you wish.
9. Click *OK* to save the addition(s) and exit the dialog box.

Step 2: Configure Printer(s) for the User

Configure the printers for the Web User in the same manner as you have previously added and configured printers. See Chapter 2 in the LLM User's Guide for more information.

Client-side Connections

Configuring Connections to the Web Listener (i-Push)

Note: The following steps detail your connection and configuring Printers on the Listener (Client) Side.

1. Press Start > Programs > Loftware Labeling > Print Server > Internet Clients > Web Listener – Interactive and press Start to Login.

If you are starting the Web Listener for the first time, the following is displayed:

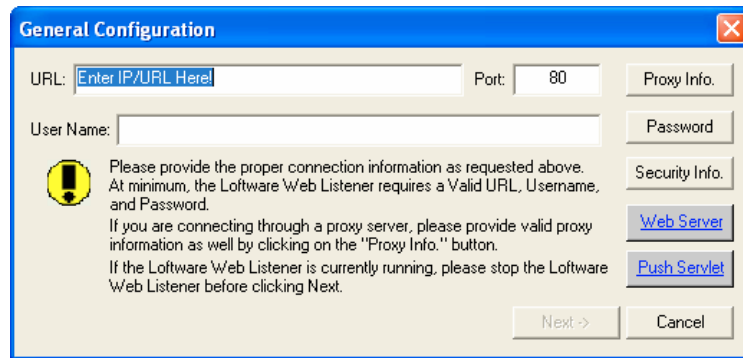


Figure 3-V: General Configuration Dialog Box

2. Enter an *IP Address* or a *URL*, such as 152.22.0.58 or www.yourcompany.com, (Port default=80). If your servlet engine is configured to use a port other than 80, enter that number.

Important Note: *The IP address or the URL is the LPS Web Server address!*

3. Enter a valid *User Name* from the LPS Users list.
4. Press the *Password* button and type the LPS User's *Password*, then re-type it in the space below to confirm; click *OK*.
5. Press the *Proxy Info* button **ONLY** if you are connecting through a Proxy Server.

If you are NOT connecting through a Proxy Server, go to Step 6.



What is a Proxy?

A Proxy is a way to connect to the Internet in a secure fashion. A Proxy is similar to a firewall, and in Loftware's case, all clients going through the proxy share this extra security measure. It is generally placed between a client application and the Internet. Proxy servers are able to improve performance for groups of users, as it saves (caches) the results of all requests for a certain amount of time, therefore recalling the information is a much faster operation. Proxy servers can also be used to filter requests, for example, a proxy server could be used to prevent employees from accessing specific Web sites.

The following is displayed:

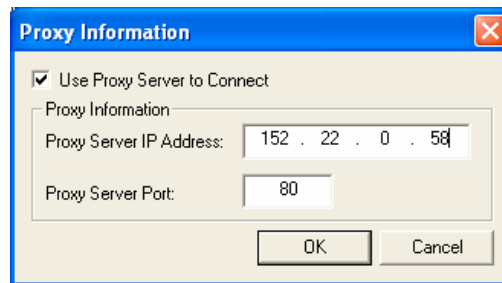


Figure 3-W: Proxy Information

- a. Click in the check box to enable the use of a Proxy Server to connect.
- b. Enter *Proxy Server IP Address* and *Port*.
- c. Click *OK*.

You are returned to the General Configuration dialog box.

6. Press the *Security Info* button.

The Web Security Configuration Utility is displayed:



Figure 3-X: Web Security Configuration

Sections of the Web Security Configuration Utility

Enable Win INet Support – This box is checked when you need to authenticate through a proxy using Microsoft's Proprietary NT/Lanman Authentication.

Note: You should NOT check this box if you plan on using the Web Listener as a Service as Win INet is not supported in Service mode.

7. Choose *Proxy User* and/or *Website User*, and complete the Security Information.

When would you choose both?? – If you are connecting through a Proxy Server AND your website has security access, you must configure both Proxy User and Web Server.

When would you choose just one?? – If you are NOT connecting through a Proxy Server, but your website has security access, configure just the Website User utility. If you are only connecting through a Proxy Server, and your Website does NOT require security access, then configure only the Proxy User utility.

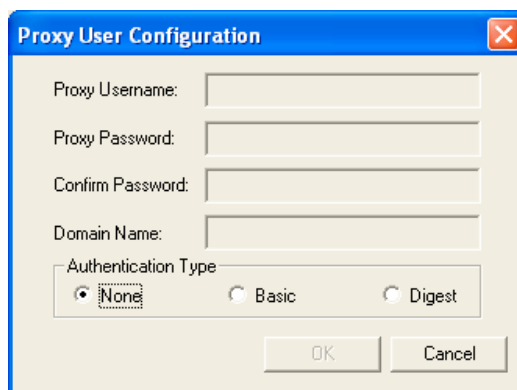


Figure 3-Y: Proxy User Configuration

Sections of the Proxy User Configuration Utility

Proxy Username - The name of the Proxy User connecting to the Web Listener.

Proxy Password - The password of the Proxy User connecting to the Web Listener.

Confirm Password - Confirm the Proxy User Password by re-typing it.

Domain Name - The name of the domain the Web Listener is connecting to.

Authentication Types Section

None – This is the default authentication type and should be used if your Proxy Server does NOT require authentication (password, etc.) to access the Internet.

Basic – This is Basic (64-bit) Authentication and should be used if your Proxy Server *does* require the use of Basic Authentication to access the Internet.

Digest – This is Digest (128-bit, MD-5) Authentication and should be used if your Proxy Server requires the use of Digest Authentication to access the Internet.

Note: Information about Authentication Types may be found on the Internet in numerous locations, including: <http://www.ietf.org/>

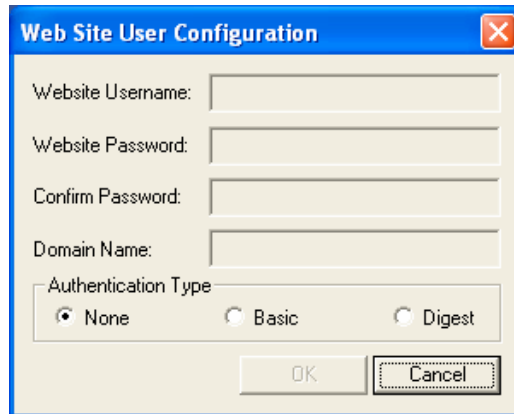


Figure 3-Z: Web Site User Configuration

Sections of the Web Site User Configuration Utility

Website Username - The name of the Website User connecting to the Web Listener.

Website Password - The password of the Website User connecting to the Web Listener.

Confirm Password - Confirm the Website User Password by re-typing.

Domain Name - The name of the domain the Web Listener is connecting to.

Authentication Types Section

None – This is the default authentication type and should be used if your Website does *not* require authentication (password, etc.) to access the Internet.

Basic – This is Basic (64-bit) Authentication and should be used if your Website *does* require the use of Basic Authentication to access the Internet.

Digest – This is Digest (128-bit, MD-5) Authentication and should be used if your Website requires the use of Digest Authentication to access the Internet.

Web Server – This hypertext link provides you with a quick way to see if the Web Server is running by opening the Web Server in your default browser. If this is not able to be opened, then the Web Server is not connected!

Software Push Servlet – This hypertext link is also a quick way to see if the Web Servlet is operational. If it is, the page is displayed in your default web browser. If this page does not open, the servlet is not turned on, and this page does not display. If the servlet is turned on, but it is not connected to the Software Print Server, the page displays with a “not connected” message.

8. When the Server Configuration information is complete, click *Next*.

The following is displayed:

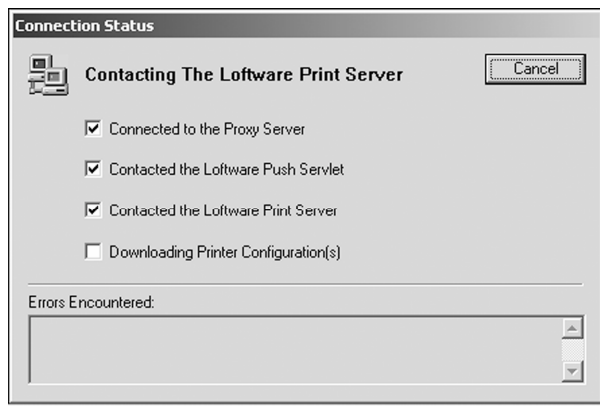


Figure 3-AA: Web Listener (i-Push) Status Screen

If the connections are completed without error, the Web Listener Configuration Utility is displayed. If the connection fails, view the error message displayed in the Errors Encountered: section, and correct the errors. A common error might be failure to connect through the Proxy Server, or typing in the wrong IP Address, either of which produces an error message.

Another common error is not having any printers configured on the Software Server. Information on adding printers to the Software Print Server is found earlier in this chapter.

Client-side Printer Configurations

At this juncture, you are required to configure your printers locally. Each section of the utility is outlined below, and they are important components in the completion of the Web Listener connection. The Web Listener does *not* work if the local printers are incorrectly configured!

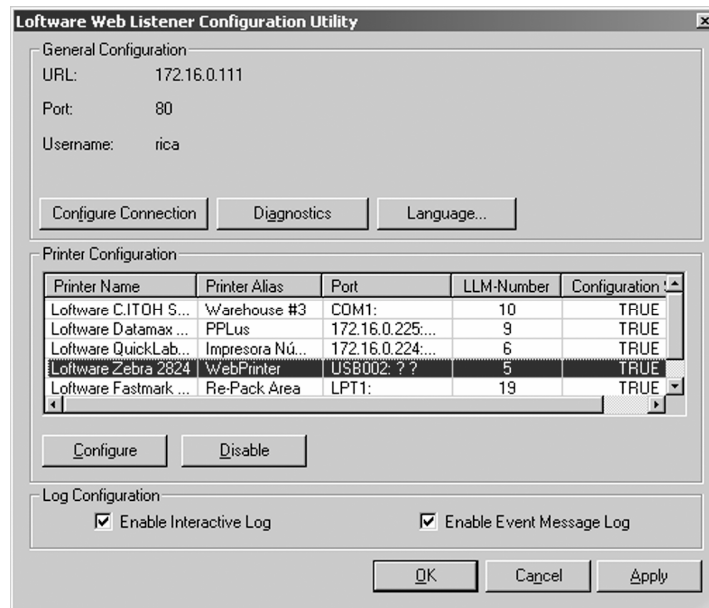


Figure 3-BB: Web Listener Configuration Utility

Sections of the Web Listener Configuration Utility

General Configuration Section

URL – The Internet address of your Web Server that is running the Push Servlet may be written as www.loftware.com, or numerically, such as 133.23.54.78.

Port – The Port address of your Web Server. The default for most servers running http is Port 80.

UserName – The Name of the logged-in Client PC as defined in the User Database on the server.

Proxy Server – URL for Proxy Server as described previously. (If using a Proxy Server)

Configure Connection – When pressed, allows you to re-configure the connection.

Diagnostics – Press this button to open a Diagnostics window for troubleshooting if you are experiencing problems with the Web Listener.



Figure 3-CC: Diagnostics Window

Test Configuration – This starts the Connection Status process.

Web Server – This hypertext link provides you with a quick way to see if the Web Server is running by opening the Web Server in your default browser. If this is not able to be opened, then the Web Server is not connected!

Loftware Push Servlet – This hypertext link is also a quick way to see if the Web Servlet is operational. If it is, the page is displayed in your default web browser. If this page does not open, the servlet is not turned on, and this page does not display. If the servlet is turned on, but it is not connected to the Loftware Print Server, the page displays with a “not connected” message.

Printer Configuration Section

This list is generated from the User Database on the LPS. Only the printers for the connected User are displayed.

Printer Name – The name of the Loftware printer to which you are connected.

Printer Alias – The Printer Alias assigned to this Printer.

Port – The Port denotes how the printer is connected (TCP/IP or Local [i.e.; COM1] Connection). This column is not populated until the port has been configured on the Listener side.

LLM-Number – The number of the Printer as listed on the Server-side.

Configuration Status – Listed as either True or False. True indicates that the printer is configured on the Listener-side; False indicates that this printer has not been configured on the Listener-side.

Configure – Opens the Printer Configuration Dialog Box to configure a printer. This is grayed out until a printer is selected.

Disable – Disables the selected printer from the Client-side Printer List. This is a helpful choice when a printer has been deleted from the Server-side Printer List. The disable button, however, does not affect the Server Side list. This is grayed out until a printer is selected.

Log Configuration Section

Enable Interactive Log – When checked, a detailed and specific log file is written to the Web Listener window. If it is left unchecked, only basic messages are displayed such as start, stop, and basic error messages. This only applies to Interactive Mode.

Enable Event Message Log – When checked, specific and detailed event information is sent to the Event Log. This log can be viewed with the Event Viewer, accessed in the following manner:

WIN2000: (Start>Settings>Control Panel>Administrative Tools>Event Viewer.)

WINNT: (Start>Programs>Administrative Tools>Event Viewer.)

Note: EVENT LOG NOT AVAILABLE ON WIN9x If your Internet Printing using Web Push Technology is successful, and no problems are occurring, then it is best to leave both boxes unchecked to save resources. If you are having problems running the web push, either interactively or as a service, then enabling these check boxes helps you to pinpoint with more accuracy where the problem is, and how to proceed.

Configuring Printers on the Client-side

1. Click on a Printer in the Printer Configuration Section; press the *Configure* button.

The Printer Configuration dialog box is displayed:

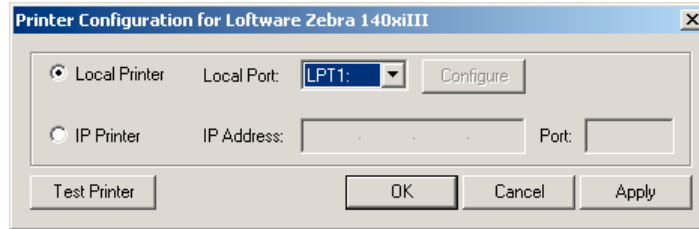


Figure 3-DD: Listener Client Printer Configuration

2. Configure a Port for the printer; choose a Local Port (i.e., COM 1, LPT 2, USB, etc.) from the drop-down list, or choose the appropriate IP address; click *Apply*.

Note: *USB Ports are ONLY listed in the Local Port list if the USB Printer is connected and is powered on. See Chapter 2 of the LLM Guide for more USB information.*

The *Test Printer* button becomes active.

3. Press the *Test Printer* button, and a small print stream is sent to the configured printer.

If successful, the printer prints the print stream, if not, then check the printer to make sure the printer has the power on, that the Port is working, etc. See Chapter 2 and Appendix B of the LLM User's Guide for more documentation on printer connections and error messages.

More Printer Configuration Information

In Interactive Mode, you can check to see what Local Ports are available for the Web Listener by choosing **Options > Check Local Ports** from the Menu Bar. A box is displayed that lists the available local ports (i.e., COM 1, LPT 2, USB, etc.) You may use one of these ports or a TCP/IP Port as shown in the Configure section. This is a diagnostics tool, and is not a necessary step.

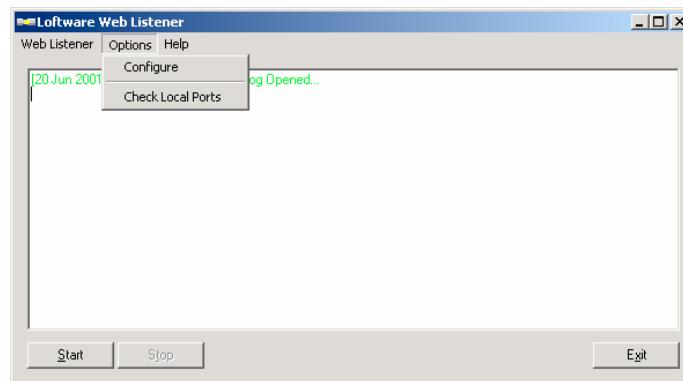


Figure 3-EE: Web Listener (i-Push), Interactive Mode.

You can also access the Web Listener Configuration Utility in Interactive Mode by clicking 'Options > Configure' on the Menu Bar.

Using Web Listener as a Service

Starting Web Listener as a Service

Start>Settings > Control Panel > Administrative Tools > Services > Loftware Web Listener

Note: Initially, Web Listener should be started interactively to configure printers, thereafter, it should be run as a service.

Using Web Listener as a Service

A service is an application that runs in the background, unseen, and processes information without user login. (See the LPS Chapter for more information on the LPS as a service.) Web Listener runs as a service, staying connected to the Web Server and the LPS with the help of the LPSPushservlet, as described previously. A file dropped to the LPS is processed by the Listener Client and printed if the configurations have been followed correctly.

The Web Listener is designed from the ground up to run as an NT Service with all applicable security. An NT service is an application that can be configured to run when the PC is booted. It runs at a lower level (ring) than ordinary programs and can not be seen or changed by the operator.

Services perform their functions without requiring the operator to log onto NT, thus providing protection from intentional or accidental change (security). If you are using NT4 Service Pack 4+ or Windows 2000, we recommend that you run the Web Listener as a service. Both the Workstation and Server editions of NT/2000 support services.

Summary of Internet Printing

This chapter has covered the newest technologies that Loftware has developed, namely the Internet Applications known as the Loftware Web Servlet, WebClient (i-Pull), Web Listener (i-Push), LPSPushservlet and Data Push (Web Push). For the latest changes, please review the online documentation at www.loftware.com.



Chapter 4 LPS Clustering

What is Clustering?

Note: Clustering is a large and complex topic. The following discussion covers clustering as it applies to maximizing the availability of the Software Print Server (LPS) in your enterprise. Some of the points made here do not necessarily apply to other “cluster aware” applications. If you have not set up cluster aware applications in the past, find someone who has to assist you.

Clustering is a feature of the Software Print Server Edition. In its most basic form, two servers are connected together with a central storage device between them as shown in the following figure. The cluster service makes the two machines appear as a single virtual server on the network. Client applications make their connections to this virtual server. If an application “goes down” due to a hardware or software failure, it is automatically restarted on the secondary server. This is referred to as a “failover.”

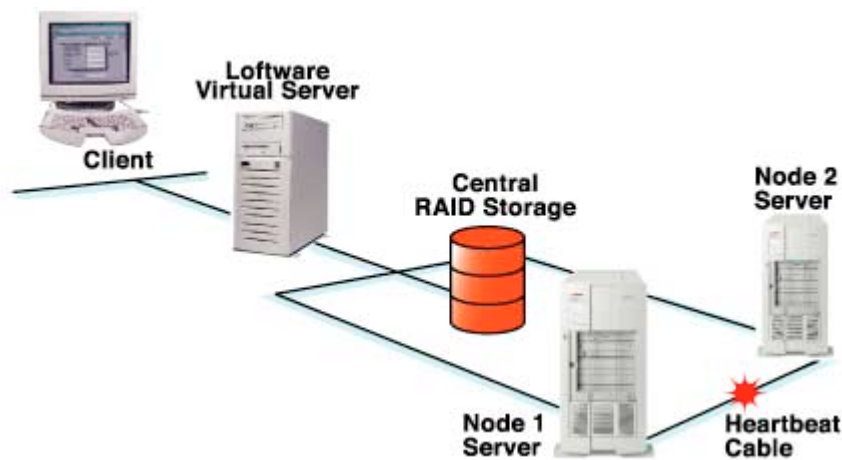


Figure 4-A: Basic Cluster Architecture (Active/Active)

At this time, Software is using the Microsoft® NT/2000 Clustering Service (MSCS) exclusively. The Microsoft® Clustering Service architecture is designed to work with standard Windows Server storage drivers interfacing with SCSI devices. The storage device, typically a RAID system, must be connected to both nodes in the cluster for failover to work properly. Shared storage devices must be formatted using NTFS.

Clustering requires special hardware, software, and expertise. You cannot simply “network” two computers together. The extra expense of hiring a knowledgeable person who is intimately familiar with clustering computers together is well worth it. Software cannot help you with the setup or support of your hardware.

Definition of Terms

When you begin to familiarize yourself with a clustered environment, some concepts and terms can be quite daunting if you do not understand what they mean. As the remainder of this document uses these terms quite frequently, they are defined here for your convenience. Some of these definitions are from the following books, which are available from Amazon.com, and are considered to be “must haves.”

“Windows NT Cluster Server Guidebook”, Prentice Hall, by David Liberton

“Windows 2000 Cluster Server Guidebook”, Prentice Hall, by David Liberton

Active/Passive Cluster - In an active/passive clustering solution, a standby server monitors a continuous signal from the active server. The standby server remains in a backup, passive mode until it recognizes that the active server has failed. It then comes online and takes control of the cluster. When the primary server comes back online, manual intervention may be necessary by the administrator to revert the systems to their original state.

Active/Active - In an active/active configuration, all servers in the cluster can run applications and act as backup servers to one another. There is no concept of a primary or standby server. All servers can dynamically assume the other’s role. The Software Print Server uses active/active.

Cluster Administrator - The Cluster Administrator is your view into the cluster and it is used often. It allows the creation of groups and resources. It is also where the LPS is started and stopped. Be extremely careful about adjusting resources that you do not understand as unpredictable results may occur. The following figure shows the Cluster Administrator.

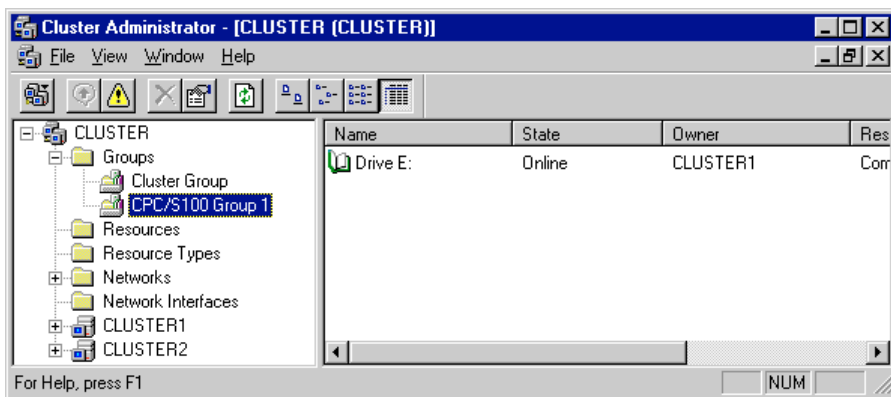


Figure 4-B: Windows Cluster Administrator

Cluster Service - A component of the cluster software that is implemented as a Windows Service. It manages all aspects of the cluster’s operation including the cluster database. Each node on a cluster runs a copy of this service.

Central Storage - As depicted in [Figure 4-A](#), there must be a central storage device between the two servers. Preferably, this should be a SCSI RAID system with at least three drives. This is where the cluster database and all Software shared files reside. Therefore, if one of the drives fails, the files are still online and available to the applications that need them, in this case, the Cluster Manager and LPS.

Client Connections - Clients connect to the LPS using the virtual server IP address, rather than a physical Windows IP address. At failover, client reconnection is necessary. The Loftware Status Client and the Notification agent reconnect automatically after the failover transition. The On Demand Print Client receives a disconnect message and has to be restarted. Any applications using the ActiveX Client Control need to trap the error event and inform the user of their options.

Failover - The process of taking one or more resources off-line on one cluster member and bringing them on-line on another.

Group - A logical organization of resources. Resources are not failed from one cluster member to another, groups are. This allows the administrator to place dependent resources into one unit, which guarantees that the same cluster member always owns a resource and all its dependencies.

Heartbeat - A message sent between cluster members to notify each other of their existence. This is done via a special hardware cable. If heartbeat messages are not received from a cluster member, it is considered to have gone off line, and all resources that it owned are failed over to the other cluster member.

Quorum Resource - The central storage drive array is usually divided into multiple partitions, one of which is designated "the quorum." The cluster services use a directory on the quorum partition to store information in the cluster database. Because it is stored centrally, this information is available to all nodes in the cluster. All partitions on central storage, including the one containing the quorum, may be used by programs that wish to store their data centrally. At bootup, only one of the machines in the cluster can "own" a specific partition in central storage.

Resource - A physical or logical entity managed by a cluster member. Resources are contained in groups. The LPS is such a resource.

Transition Time - The time necessary to relocate (failover) resources and groups to the secondary server. Depending on the number of resources that need to be moved, transition time can range from 10 to 60 seconds.

RAID - This term means 'Redundant Array of Independent Disks'. RAID is a storage mechanism that uses two or more disks to provide one logical disk that supports varying levels of performance improvements and fault tolerance. The Cluster Service only supports RAID on the shared SCSI bus. As described later, the shared Loftware files reside on the central RAID storage, as displayed in Figure 4-A.

Shared Nothing - An implementation of a cluster that does not allow resources to be accessed simultaneously by multiple cluster members. The Loftware Print Server runs in an Active/Active, Shared Nothing Cluster.

Virtual Server - A collection of resources that supply the appearance of a Windows NT/2000 server to clients. This is depicted in Figure 4-A. Rather than associating LPS with a physical Windows NT/2000 Server, the Clustering Service virtual server technology allows you to place the LPS into a virtual server. At a bare minimum, a physical disk resource, a network name resource, and an IP address resource are required to bring a virtual server online. Depending on the function of your virtual server, other resources may be required. The Loftware Virtual Server requires several additional resources, some of which are automatically created for you during the installation procedure (for more information, please refer to "Installing the Loftware Print Server on a Cluster").

Client applications connect to the virtual server, which invisibly re-route the connection to the node that is currently hosting it. Client applications should **never** connect directly to a node. The virtual

server's name and IP address are independent of the physical node name and IP addresses on which they run. The names and IP addresses of the physical Windows NT/2000 Servers and the LPS virtual server *must* be different. Regardless of which physical Windows NT/2000 server in the cluster controls the LPS, the virtual server name and IP address remains the same.

Loftware Print Server (LPS) Failover Support

The Cluster-aware version of the Loftware Print Server is only available in the LPS Premier Edition for which a backup license has been purchased. Failover Support provides the ability to make the LPS operate as a single virtual server when installed on a cluster.

Client connections see only the single LPS and are unaware of which physical Windows Server node is processing their request. As a virtual server, LPS is managed through Cluster Administrator from either node of the cluster. The Clustering Service monitors the status of the primary and secondary nodes, the LPS application, and shared disk resources. Upon failure of the primary node, services fail over automatically to the secondary node. The secondary node becomes the primary node and vice-versa. Because the LPS runs in active/active mode, there is no need to fail the service back once the original problem that caused the failover is resolved.

Failover Criteria - Running an application in a cluster increases the availability of your application, in this case, the LPS. Although rare, there are cases where the LPS could become unavailable and no failover occurs. You must also understand that there is a transition time (defined above) associated with failovers. Depending on the nature of the failure and what the LPS is processing at the time, printing stops and new requests may not be buffered. Expect to lose LPS availability from ten to sixty seconds during the failover transition.

Retry Before Failover - Each resource is configured to retry 3 times before initiating the failover sequence. This is a double-edged sword because, on the one hand, the cluster manager might be able to successfully restart the resource, therefore avoiding a failover and eliminating the transition time. On the other hand, it takes time to perform these retries, which, if they fail each time, increases the transition time. When the setup program creates the resources, it sets them to retry three times. You may adjust the retry interval by viewing the properties of the resource. It is not recommended that a value less than two be used.

Conditions Causing Failover - The following list defines some scenarios that cause a failover. These are the ones that we have experienced and tested; there may be others.

- Any hardware failure on the quorum node that terminates the heartbeat causes a failover.
- Although rare, if the Loftware license key fails, a failover occurs. The application event log reflects the reason for the failover.
- If the LPS "goes down" and cannot be restarted, a failover occurs.
- If no printers are defined, a failover occurs. Because printer configurations are shared, the LPS fails on the secondary node as well.
- Any other case where the LPS shuts itself down due to lack of resources causes a failover.

Conditions that DO NOT cause failover - Unfortunately, conditions can occur that can make the application unavailable, but do not result in failover. This is rare, but it can happen.

- Any hardware failure that does not result in the loss of the heartbeat could make the LPS unavailable. Printing may or may not stop depending on the nature of the failure.
- If the application stops responding to print requests, but remains running in memory, a failover does not occur. Unless another application compromises one of the files that LPS relies on, this should never happen, but it **is** possible.

Note: If a drive on the central storage fails, a failover does not occur, but the LPS is still available to service print requests. This is the benefit of RAID storage. This may not be true if your central store is not using RAID.

Manually Initiating a Failover - For testing purposes, the need arises to cause a failover to make sure the desired result is achieved. There are several ways to do this, two of which are mentioned here.

1. Power down the node that owns the quorum. This terminates the heartbeat signal and causes a failover.
2. Right click the Loftware Print server resource and choose 'Initiate Failover'. The cluster service tries to restart the LPS on the same node a number of times before actually failing over. Therefore, you must initiate failover 3 or 4 times, depending on the retry interval settings.

Shared Files and Central Storage

When the LPS is installed on a cluster, its files are divided between the target NT Server node and the central storage. Executables and their dll files are installed on the NT server, while label directories, image files, layouts, etc. are stored in central storage. In other words, the files that are needed to run the LPS are installed to the server, and the files needed to print your labels are stored in central storage. This is why you must install once to each node. Printer setups and preferences are stored in the registry that is the topic of the next section.

The Mirrored Registry

One of the resources created by the setup program is called the "Loftware Mirror Manager." This resource mirrors two registry keys that are used by Loftware. The keys are "**HKEY_LOCAL_MACHINE\SOFTWARE\Loftware, Inc.**" and "**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WatchDogNT**". In the event of a failover, any printers that you have configured or preferences that you have set are copied to the registry on the other node.

All Loftware applications require that the mirror manager be running. Furthermore, even if the mirror manager is running, you cannot invoke any Loftware applications on the node that does not own the quorum. This guarantees that any changes that you make are picked up by the mirror manager and are made available to the other node in the event of a failover. A warning message is displayed if you try to invoke an application that is not covered by the mirror manager.

Note: The registry keys mentioned above are only copied to the other node during a failover. Do not expect the keys to be "in sync" if a failover has not occurred. The registry editor should be closed when configuring your system. If it is open, your changes ARE NOT mirrored!

Installing the Loftware Print Server on a Cluster

Installing the LPS to a cluster is easy to do providing that the hardware is setup correctly and instructions are followed. The setup program automatically detects that you are installing to a node on the cluster and guides you through the appropriate steps.

At this point, we must mention again how important it is to have a person with a concrete knowledge of clustering complete your installation and testing!

Step 1 – Verify the Hardware

The hardware must be setup properly in order for the installation and subsequent operation of the Loftware Print Server to be successful. When you look at your hardware, it should resemble the setup depicted in [Figure 4-A](#).

- Each node should have two network (NIC) cards.
- A heartbeat cable should connect the two nodes.
- A central RAID storage device should be connected between the nodes.
- Two Loftware Premier License keys are present.

Note: It is beyond the scope of this chapter to comment too deeply on the hardware requirements. Your best source of information when it comes to this is the books mentioned above and personnel that have implemented clustered solutions in the past.

Step 2 – Verify the Software

- The operating system must be Windows NT or 2000 Server, Premier Edition.
- Microsoft MSCS Clustering services must be installed and operational.
- You must have a copy of Loftware Premier Edition, **Version 5.4** or higher
- A backup license for Loftware Premier Edition (second license key).

Step 3 – Understanding what you are doing

There are some rules that absolutely have to be followed. Setup reminds you of these rules with the information placard displayed in the following figure.

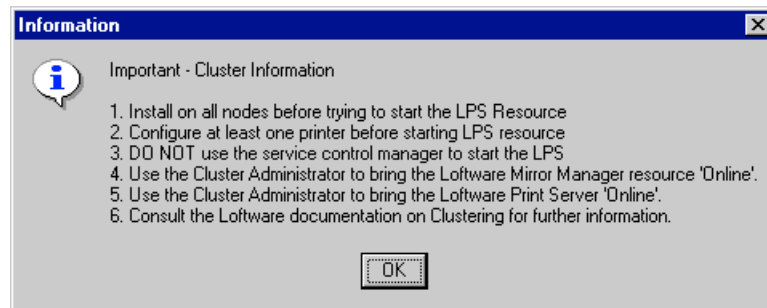


Figure 4-C: Rules that MUST be followed!

The rules listed above take precedence over any rules, guidelines, or procedures that have been discussed in other chapters of this manual! We have made all of these mistakes in the Software lab and we urge you not to do the same!

Note: For the purposes of this conversation, let's assume that there are two nodes on your cluster, named 'CLUSTER 1' and 'CLUSTER 2'. CLUSTER 1 is the current owner of the central drive resource.

- **You must install on all nodes before attempting to start the LPS resource.** -
The reason for this is as follows: Let us say that you have installed on CLUSTER 2 and attempt to bring the Software Print Server resource online. It would immediately fail because the Cluster Administrator would try to launch the LPS on CLUSTER 1 because it is the central drive resource owner, in this case the quorum owner. This would cause a failover of the quorum resource to CLUSTER 2, which may not be desirable to you. If the reverse was true and you installed to CLUSTER 1, the initial attempt to bring it online would succeed, but you would be left with a false sense of security since the LPS would fail to start on CLUSTER 2 during a failover situation. Avoid this scenario by always installing to both nodes before attempting to bring the Software Print Server resource online.
- **You must configure at least one printer before bringing the LPS online.** -
The LPS exits if you do not have any configured printers. The problem here is that this causes a failover to the other node that does not have configured printers either. The Cluster administrator fails the group back and forth until it finally gives up.
Avoid this by configuring at least one printer before attempting to start the resource.
- **DO NOT use the operating system's Service Control Manager to start the Software Print Server service** (contrary to instructions given earlier in this manual). Doing so nullifies any failover capabilities and could cause unpredictable behavior. Furthermore, DO NOT set the LPS service to automatic start in the service control manager. The Cluster Manager takes care of this for you. Think of the Cluster Administrator as your new service control manager.
- **Always use the Cluster administrator to bring the Software Mirror Manager and the Software Print Server resource online/offline.** The Cluster Service remembers the state that these resources are in for later use if/when the PC is rebooted. Once you are up and running properly, you need not log into NT. The Cluster Service auto-starts the LPS and failover whenever conditions warrant it.

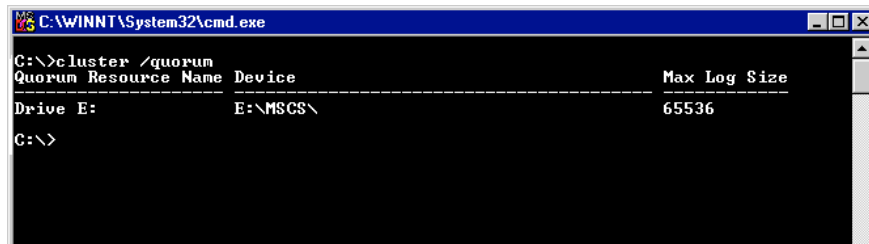
Step 4 – Log into the system

The NT/2000 operating systems require a password for the Account under which the LPS runs. This account must belong to the Local Administrators' group. If you are not logged onto the account under which the LPS is running, log in under the appropriate account before running the setup program. If you do not have a suitable login account, have your network administrator create one for you. Remember, the account must have administrative privileges, and both nodes should run under the same account.

Step 5 - Determine which group contains the Quorum Resource

The central storage drive array is usually divided into multiple partitions, one of which is designated "the quorum." The cluster services use a directory tree on the quorum drive to store information in the cluster database.

1. Go to a command window (**Start->Run...cmd**). Type in **cluster /quorum** as shown in Figure 4-D. The quorum resource name, 'Drive E' along with its directory tree, 'E:\MSCS', is reported.
2. Open the Cluster Administrator and expand the Groups section
3. Click through the groups until you find the one that contains the quorum drive resource name (Drive E in this case).
4. Figure 4-E shows that the group containing the quorum for this example is '**CPC/S 100 Group1**'



```
C:\WINNT\System32\cmd.exe
C:\>cluster /quorum
Quorum Resource Name Device Max Log Size
-----
Drive E: E:\MSCS\ 65536
C:\>
```

Figure 4-D: Using the console to determine the quorum drive

Software Version 5.4 requires that you install the Software Resources to the quorum group. You may go into production this way, or manually create a different group and move our resources to the new group. Be sure that the new group has a drive resource on the central store, which you must manually create.

Software Version 5.5 and later - allows you to install to any group that you choose. You must manually create the group and be sure that it contains a drive resource in the central store before you may continue.

Step 6 – Determine which node 'owns' the quorum resource

Select the Quorum Group as described above. When selected, the owner is listed in the 'Owner' column. The owner for our example is 'CLUSTER1'.

Note: Select your resource group and note the owner of drive resource with Software Ver. 5.5+.

Step 7 – Create the Software File Share

Before running setup, you must create the 'loftware\$' file share resource. Create this in the quorum group if you have Loftware Ver. 5.4. Later versions allow you to create the 'loftware\$' resource in any group providing that it contains a drive resource to central storage. In either case, the 'loftware\$' resource must depend on the drive resource. Figure 4-E shows the quorum group which contains the loftware\$ share.

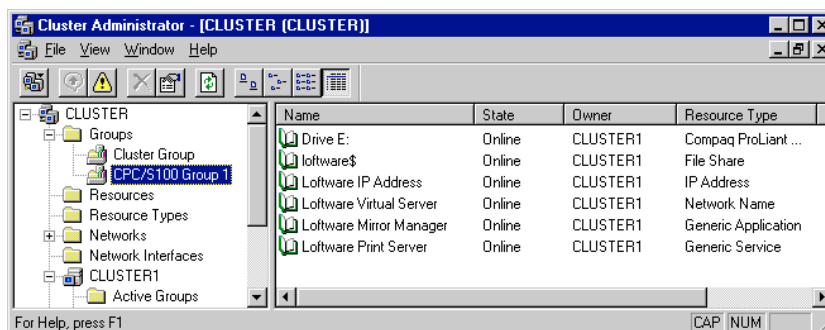


Figure 4-E: Cluster Administrator showing the loftware\$ share

1. Determine ownership:
 - a. **Loftware Version 5.4:** Determine which node (PC) owns the quorum resource. This is explained in Step 6.
 - b. **Loftware Version 5.5** or above: Look for the owner of the drive resource in the group that you have created for the Loftware resources. This is explained in Step 6.
2. Log onto the PC that owns the resource specified in the previous step.
3. Launch Windows Explorer and navigate to the central store ('Drive E' in this example)
4. Create a directory on the central store. The example below uses the name 'LoftwareCentral', but you can name it anything you want. The figure below shows this directory as seen from CLUSTER 1, the drive resource owner. The drive is not visible to CLUSTER 2 unless a UNC path is used from **Start->Run**.

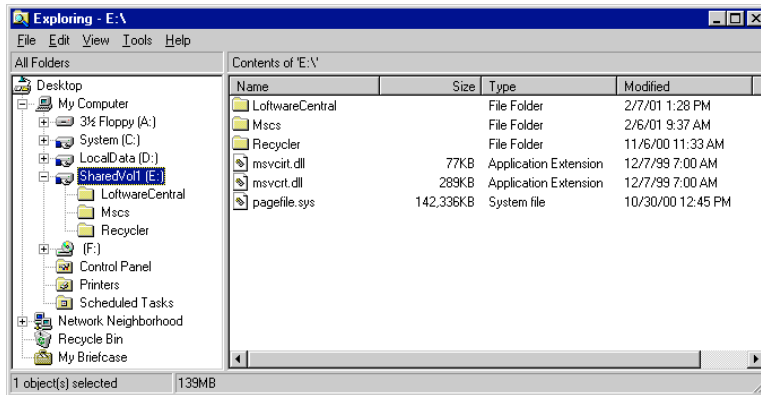


Figure 4-F: Loftware directory on the shared central drive

5. Invoke the Cluster Administrator on CLUSTER 1
(**Start >Programs >Administrative Tools**)
6. Choose the group to which you wish to install.
(Version 5.4 requires that you install to the group containing the quorum resource).
7. Check that the group has a drive resource on the central store. If it does not, you must create a drive resource that references one of the petitions on the central store.
8. Right-click in the group and choose **New >Resource**.
The screen shown in the next figure is displayed:

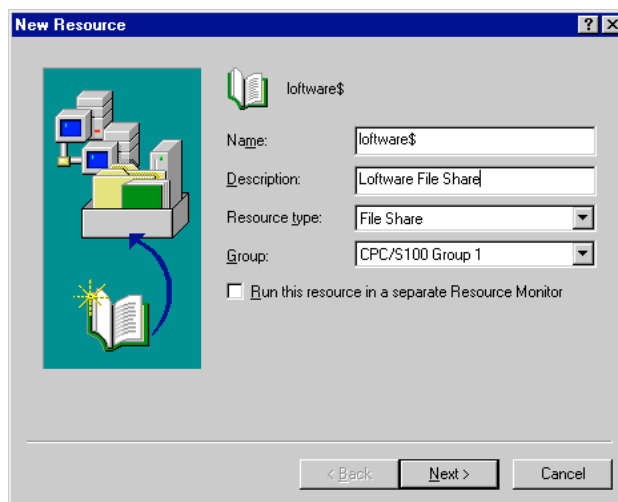


Figure 4-G: First Screen for Creating a File Share Resource

The name of the resource must be **loftware\$**

9. Type 'Loftware File Share' for the Description.
10. Choose **File Share** for Resource type as shown in the figure above.

The Group already is selected on the quorum if you invoked the wizard from the quorum. See the note at the beginning of this section for information on determining which group is the quorum.

11. Click **Next**. The screen shown in the following figure is displayed:

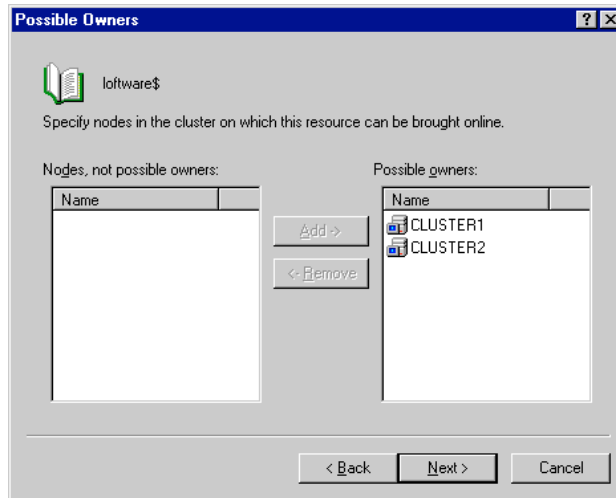


Figure 4-H: Both Nodes are possible owners

12. Check that both nodes are included under the **Possible owners** column.
13. Click **Next**. The following figure is displayed:

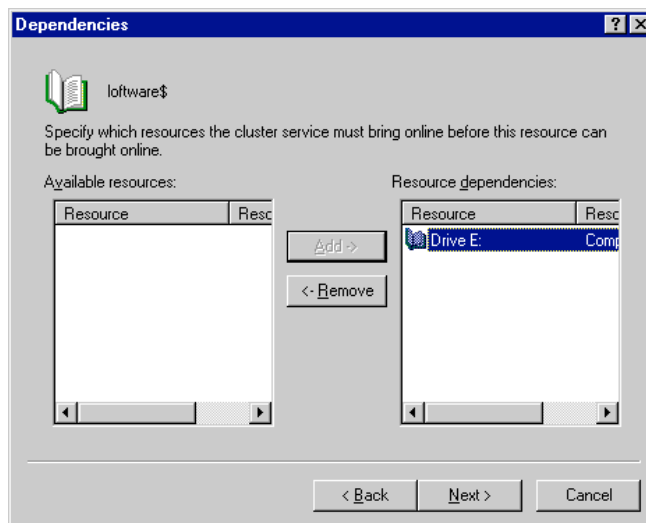


Figure 4-I: The File Share must be dependent on the mapped drive.

14. Add the central drive resource name to the Resource Dependencies column by clicking on it and choosing **Add**, as shown in the preceding figure. **This causes the cluster administrator to bring the drive up before the share, which is a must!**
15. Click **Next**.

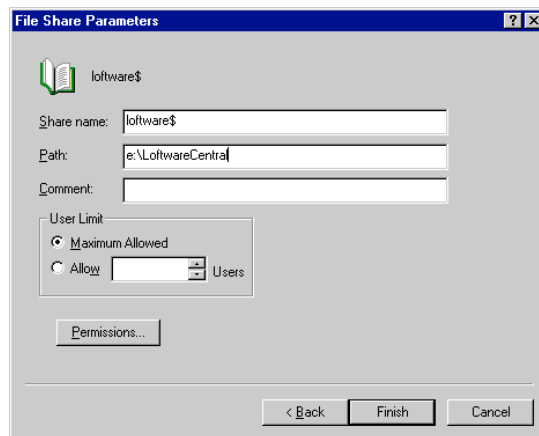


Figure 4-J: Path to the share

The share name must be `loftware$`. DO NOT use a different name

16. Describe the file share using the mapped drive path, in this case '**e:\LoftwareCentral**'. Do not use the UNC path in this case! This is shown in the figure above.

17. Set permissions default to **All, Full Control**. You may change these here if you wish.
(Make sure that you do not cut off any of your Clients by denying permissions to the share.)
18. Click **Finish**, and your loftware\$ share is displayed in the quorum group.
19. Right-click the loftware\$ resource and choose **Bring Online**.

Note: If you do not bring the loftware\$ resource “online,” the rest of the setup process fails.

Step 8 - Installing the Hardware License key

If you have purchased the Loftware Premier Edition with backup support, you should have received two hardware license keys. Plug one of them into the port of Node 1 and the other into Node 2. It does not matter which key plugs into which node. The LPS does NOT run at all on a cluster machine without the Premier key. Refer to Chapter 1 of the LLM User’s Guide for more information on key installation.

Step 9 – Launching Setup in Cluster Mode

Note: If you currently have Loftware Version 5.3 or below on your system, you must uninstall Loftware before proceeding with this step, or the proper clustering setup dialog boxes are not displayed.

Note: The setup steps outlined here do not cover every aspect of the setup program, only the parts particular to clustering. If you see a screen that is not documented here that you do not understand, refer to the installation chapter for more information.

Place the Loftware CD in the drive and wait for autorun to start. If, after waiting at least 15 seconds, it does not start, navigate to the root of the CD and double-click **setup.exe**. After choosing **Next** from the welcome screen, the following screen is displayed if clustering services are available. If the following screen is not displayed, your cluster is not set up correctly, or you did not uninstall your previous Loftware version. Please note that it does not matter what machine you install to first. The important thing is that you remember to install on both machines!

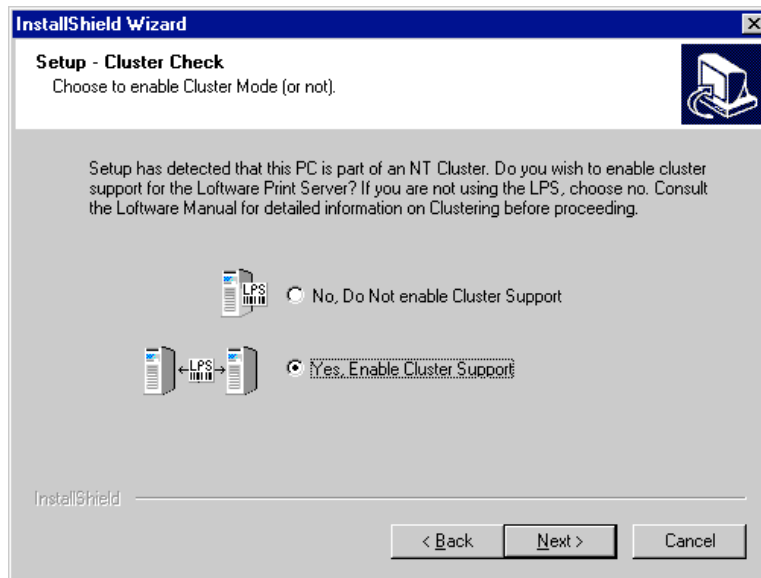


Figure 4-K: Prompting for a Cluster Install

Be sure to choose **Yes, Enable Cluster Support** before pressing **Next**. If you choose No, the Loftware programs are installed in normal mode and no clustering support is realized.

Step 10 – Creating the Loftware Cluster Resources

After you navigate through some informational screens, setup displays the screen shown in the following figure. This information supplied is used to set up the resources.

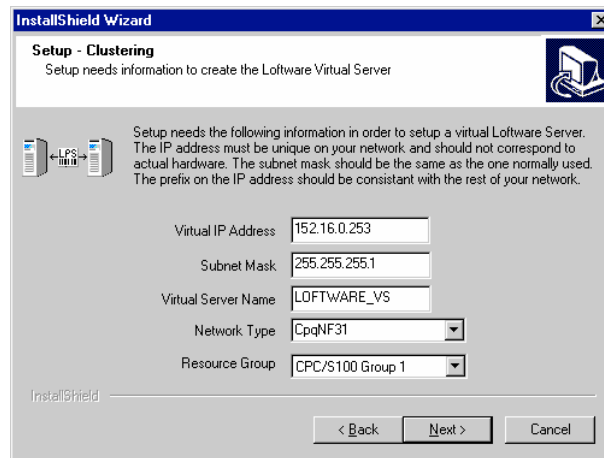


Figure 4-L: Information for creating Loftware Cluster Resources

Virtual IP Address - This is the TCP/IP address of the Loftware Virtual Server. The prefix should be consistent with the rest of your network, but the last number should be unique. A network administrator has to assign this number to you. It must be unique and not correspond to any actual hardware on your subnet. The LPS client programs rely on this address to connect.

Subnet Mask - This mask should be the same as that used on your local subnet. Again, have your network administrator assign this to you. The subnet mask and virtual IP address are used by setup to create the *Loftware IP Address* resource as depicted in the next figure.

Note: If you are not sure about the IP Address, try to ping it. If you do not receive a response, chances are that it is unique. Ensure that in the future, no one tries to use this address. If the address is used elsewhere on the network, unpredictable results occur!

Virtual Server Name - This is the network name of the virtual server. The default is “LOFTWARE_VS”, but you can name it anything that you want. This name, along with the “loftware\$” share forms the basis for the virtual UNC path to central storage. This path, by default is \\LOFTWARE_VS\LOFTWARE\$. Setup uses this setting to create the *Loftware Virtual Server* resource as depicted in the next figure.

Network Type - There are two network names in a typical cluster. One refers to the network connection between the two machines (failover network); the other is the normal connection to the company network. Choose the normal connection to the company network in the drop-down list. Different cluster setups use different naming conventions for the network type. Some name the normal network connection something like “PUBLIC.” **You must know which one to choose.** Unpredictable results occur if you make the incorrect choice. Setup uses the choice here to build the *Loftware IP Address* resource.

Resource Group – If you have Loftware Version 5.5 or greater, choose the group to which you wish the Loftware resources to be installed. Version 5.4 requires that they are installed to the quorum group itself. As noted earlier, you may move our resources to a different group after installing. Just be sure that you move all of them. The shared drive resource is not created by setup; it must be there before you launch setup.

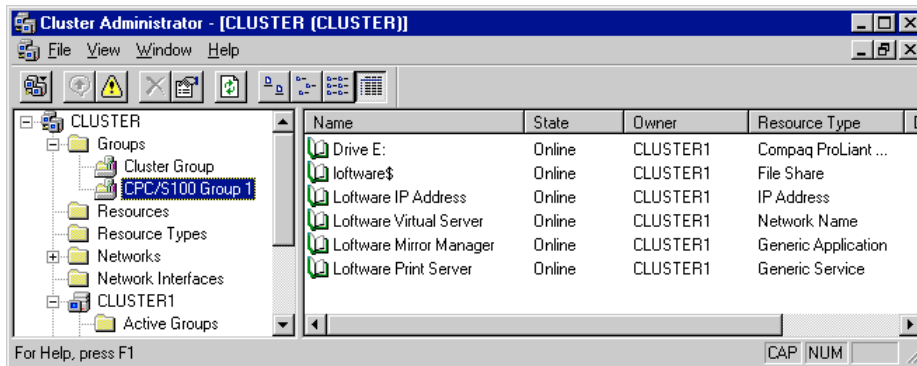


Figure 4-M: Cluster Administrator showing the Quorum group with the Loftware Resources

Note: Unpredictable behavior occurs if any resource names are changed after setup completes.

Step 11 – Configure the LPS

After a few more screens, the screen shown in the following figure is displayed. If setup asks you to reboot, the screen is not displayed until the post boot initialization sequence. Because the LPS runs as an NT service, you must supply a password that belongs to the local administrator's group. Check the User ID displayed in this dialog box. This is the account under which the LPS runs. If it is incorrect, press 'Cancel', and go back to Step 1.

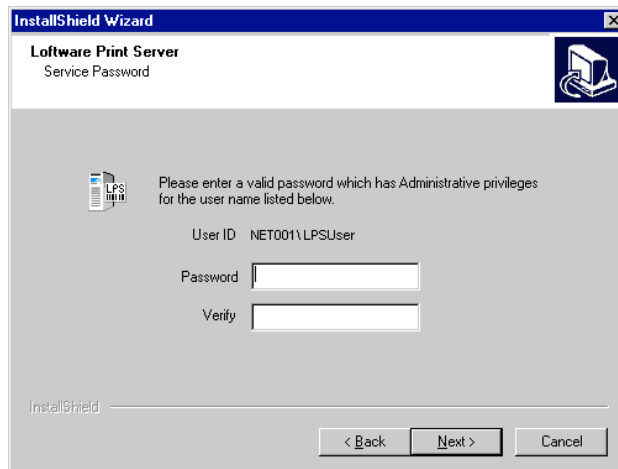


Figure 4-N: Prompting for the Password

Step 12 – Install to 2nd Cluster

Start again with Step 9, and install to the 2nd Node.

Step 13 – Preparing and Testing your System

After having successfully installed your system on both nodes, it is time to verify that your system works properly. Use the following list as a guide:

1. Make sure you have successfully installed on both nodes.
2. Invoke the Cluster Administrator on either node.
3. Expand the group containing the Loftware resources and note which node owns it.
4. The loftware\$, Loftware IP Address, and Loftware Virtual Server should all be online. If they are not, review this document starting from the top and correct any mistakes that you may have made. Right click them and try to bring them online. If they fail, you may have to call the Loftware technical support department.
5. Right click on the Loftware Mirror Manager resource and bring it online.
6. Invoke Label Design mode on the node that owns the quorum.
7. Configure at least one printer.
(Refer to Chapter 2 in the LLM User's Guide for information).
8. Design a simple test label with one or two variable fields.
(Refer to Chapter 3 in the LLM User's Guide for information)
9. Test Print your label from Design Mode, and upon successful printing, exit Design Mode.
10. Right Click on the Loftware Print Server resource and bring it online.
11. Using a text editor, such as Notepad, create a .pas file that prints your label.
12. Copy your file to the wddrop (default) directory on the central store.

Your label should print. If it does not, be sure to review Chapter 2 in the LLM User's Guide, and Chapter 1 in this User's Guide.

13. Initiate a failover by either shutting down the primary node or right clicking the Loftware Print server resource and choosing '**Initiate Failover**'. The cluster service tries to restart the LPS on the same node 4 times before actually failing over.
(Remember, you must initiate failover 4 times).
14. Drop your pass file to the wddrop directory again after the failover is complete. It should print the same way as it did on the other node.

Helpful Things to Remember

ODBC Data Sources - If you are using an ODBC datasource, do not put your database on one of the cluster nodes. If it is not a host type of database, the file should be placed in the central store. The ODBC Driver and datasource for the database must be set up on each node.

Windows Drivers and Spoolers - If your printer requires that a driver be installed, you must install the driver on both nodes of the cluster. Be sure to use the same driver name and settings on both nodes. If you are using Loftware Native Drivers with TCP/IP addresses, your printers only need to be configured on one of the nodes, the quorum. All settings transfer to the other node at the first failover.

Configuring your System to Auto-start

Note: As explained in a previous section, do not use the service control manager to set the LPS to autostart. Doing so nullifies cluster support and may cause unpredictable behavior.

When your system is completely set up and you are ready to go live, bring all Loftware resources online and log off the system. The LPS service serves all requests coming from your applications and other Loftware client modules without having anyone logged in. If one or both machines are rebooted, the cluster service automatically starts the LPS without the need for logging in. If this is **not** the desired behavior, simply take the LPS resource off line before logging out or shutting down. In other words, the cluster service returns all resources to their previous state on bootup.

Upgrading, Repairing, and Uninstalling

Things to remember before attempting to modify your installation:

- You must take the LPS and the Mirror Manager offline before performing an upgrade, repair, or uninstall.
- Remember that the LPS is offline during this process and print requests are denied.
- You must perform the same process on all nodes before attempting to restart the LPS
- Uninstalling removes all Loftware programs and resources, except the loftware\$ share. Any files that were installed or created in central storage remain. Use Windows Explorer to delete them if this is what you want.

Remote Administration of a Windows 2000 Cluster Server

From a network administrator's point of view, one of the most useful features of Windows 2000 Server is the introduction of terminal services. While Windows NT 4.0 had a special version of the operating system named Windows NT Terminal Server, all versions of 2000 Server come with terminal services available, which can be enabled to run in either Application mode or Administration mode. By default, the Terminal Service is enabled to run in Administration mode, unless additional steps (as outlined by Microsoft) are taken to set the server's Terminal Service to run as an application server.

Loftware has designed the latest versions of the Loftware Print Server to take every advantage of this modern technology, whether you have initialized Terminal Services in administration or application mode.

Terminal Services technology is also available for use with a Microsoft Cluster Server (MSCS). There are, however, some important things to consider when using Terminal Services with Microsoft Clustering Technology.

- Loftware strongly discourages remote installations/upgrades of the LPS. All installs/upgrades need to be performed locally on the server.
- Remote administration of the LPS, once installed, is supported, and gives you the ability to remotely add/remove Loftware printers, design labels, and administer Loftware users.

- Remote administration of a Microsoft Cluster is complicated by the many IP addresses which may be hosted by a Microsoft Cluster. In addition to the physical IP addresses of the actual cluster nodes, every virtual IP address hosted by the clusters creates a potential entry point for remote administration. For example, a newly installed cluster, created from two nodes, has three IP addresses that are able to gain control of the cluster; one IP address for each node and the IP address used for quorum administration. When you remotely connect to a virtual IP address or a virtual network name hosted on the cluster, you are really remotely connecting to the node that currently hosts that virtual network server.
- When remotely administrating the LPS, Loftware recommends you use the IP address or network name of the Loftware Virtual Server to terminal into the cluster. This guarantees that you always connect to the node that is currently hosting the LPS virtual server. The actual nodes of the cluster rarely need to be used for anything besides local administration of the cluster. Client connections should *always* be directed to a virtual server.

More information on virtual servers can be found elsewhere in this chapter.

Client Connections to the LPS Virtual Server

As stated previously, the two nodes of the cluster are transparent to clients that are connecting to the LPS. The clients connect to the virtual server. If there is more than one LPS virtual server or a combination of LPS servers on your subnet, the Loftware Clients displays the arbitration dialog box shown in the figure below:



Figure 4-O: Server Arbitration List as seen from a client

The figure above shows all of the LPS servers that are currently running on your local subnet with their versions and IP addresses. If you have only one installation of LPS, there is no need for arbitration and the client program automatically connects to the one it finds. Also, notice the 'LOFTWARE_VS' virtual server name. Even though there are actually two installations of LPS on the cluster, only the virtual name is available for connection purposes.

There are several ways to connect to the LPS, each of which is documented in other chapters of this manual. Each connection type may behave slightly differently when a failover occurs.

- **Loftware Notification Agent** – Tries to reconnect every 30 seconds after the connection is lost. Reconnect is automatic and usually occurs within 60 seconds of the beginning of transition time.
- **Loftware Thin On-Demand Print Client** – A message box is displayed which informs the operator that the connection to the server has been lost. It does not know if this due to a failover. If it is a failover situation, the operator must exit the application, wait 30 to 60 seconds, and try again.
- **Loftware Status Agent** – Tries to reconnect every 30 seconds after the connection is lost. Reconnect is automatic and usually occurs within 60 seconds of the beginning of transition time.
- **Loftware Thin Client ActiveX Control** – Many of the methods in the ActiveX control fail during the transition period. Be sure to trap the error event as shown in the ActiveX Client Control chapter.
- **File Interface from a host or PC application** - Writing the pass file to the drop directory fails if it is attempted during the transition time. Be sure to take appropriate steps in your application to trap this error and handle it.
- **Socket Interface from a host, PC or Palm application** - The socket connection is terminated at the start of the transition period. Your application must recognize this and either inform the user, or try to reconnect.

Troubleshooting Loftware Print Server Clustering

Problem: Not being able to install, and getting a file error.

- Check to see if loftware\$ is online. If it is not, bring it online, and try again.

Problem: Failing, and not able to start the LPS.

- Check the Event Log, which has a wealth of information.
Example: You may look in the log and see “No Configured Printers.”

Problem: The LPS is not responding, and failover is not occurring.

- When this happens, the Cluster Service is still “seeing” the LPS as running. In this case, you need to double-check your LPS configurations and check for file errors in the Event Viewer.

Chapter 5 ActiveX Client Control

ActiveX Client Control Overview

Programmers developing in 32-bit languages supporting ActiveX Controls can easily interface Loftware's bar code printing modules directly into their own applications. Loftware's Client Control diminishes the level of knowledge and expertise required to connect and print to stand-alone and networked bar code label printers. 32-bit development languages, such as Visual Basic, Delphi, Access, Power Builder, Visual C++, etc., can utilize this innovative technology.

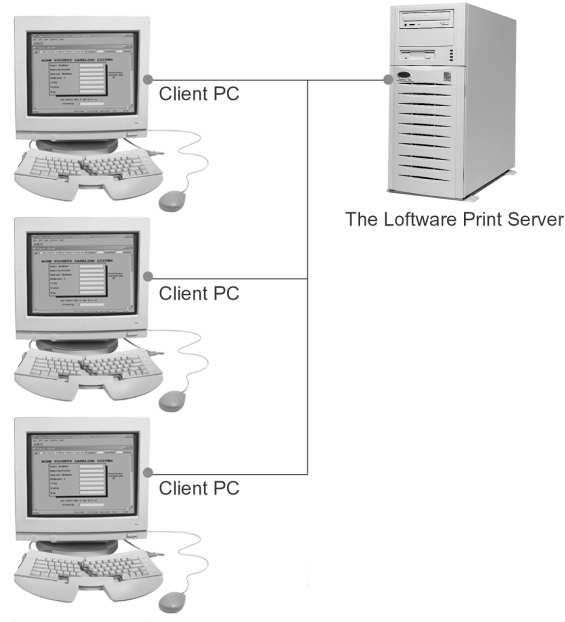


Figure 5-A: PCs using one of the Client Controls to offload print jobs to the LPS

This chapter documents the Loftware ActiveX Client Control. This control 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 "Client Control" because it acts as a client 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.

As long as you do not initialize the control by using any of the methods, or the "isRunning" Property, it does not look for the Loftware Print Server. This allows you to embed the control in your program and only use it when it is turned on either by the user or programmatically.

Beginning with Software Version 7.1, the ActiveX Client Control has the functionality to be able to connect to the LPS through a socket connection, similar to our On-Demand Print Client and our Status Client. If you are using an older version of the ActiveX Client Control with a custom application, you can upgrade to the latest version of ActiveX, which is backwards compatible. In addition, your custom application can be converted to use the new socket connection with minimal modifications. To set up your application using the ActiveX Client Control to connect to the LPS through sockets, review information found under Login Method and under ClientIniPath Property

Installation and Use of ActiveX Client Control

Installing ActiveX Client Control

1. Install the Software Print Server (LPS). Refer to Chapter 1 for instructions on how to do this. Please note that the LPS can be installed on any NT/2000/XP computer anywhere on your LAN or WAN.
2. Perform a client install of the ActiveX Control as described in this chapter.
3. Configure all connected printers (using Label Design Mode) on the Server as described in Chapter 2 of the LLM User's Guide.
4. Design the necessary label formats (using Label Design Mode) on the server.

You may also design them on any client PC and either share the files on the server or copy the files to the server when done. The **Options | File Locations** menu in label design mode allows you to set network locations for your design files. This allows you to design on a client PC while your labels are saved to the PC specified, usually the LPS PC.

5. Verify that the LPS is working by building a simple label and manually creating a .pas file for it. Copy your pas file to the directory that LPS is scanning and your label should print. If it does not, refer to Chapter 1 for ideas on troubleshooting.

Using ActiveX Client Control

1. Install the ActiveX Client Control on a client PC.
2. Load a sample program, learn, and study it. (This saves you time in the long run.)
Sample programs are installed with the LPS. They are also present in a subdirectory called sample programs.
3. Create a project in a programming tool that can utilize ActiveX Controls.
4. Add the Software ActiveX Client Control to the new project
5. Initialize the control by setting the PrinterPath, DropDirectory, .LabelsPath, and LayoutPath properties.
6. Choose a label by invoking the SetLabelName Method.
7. Set field data by using the SetData Method.
8. Print the label format directly by using the PrintJob Method.

Note: If your label does not print, try stopping the LPS scanning process to see if the control writes a .pas file to the directory specified with the DropDirectory Property.

Helpful Hints and Common Mistakes

Note: 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 ClientX methods throw critical errors during the failover transition. See the Error Event section at the end of this chapter for more information.

- Make sure that any label files created with versions of Loftware less than 3.5.2.58 are re-saved. The control uses a list inside the label format that is not in there until this version. Error Event #26503 is thrown if the label is saved with an old version.
- Do not set any Path properties of the control until you have a valid LPS installation. The control throws ErrorEvents if it cannot find the paths that you set.
- Make sure that you have some labels designed and printers configured on the LPS 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.
- Verify that the LPS is working by building a simple label and manually creating a .pas file for it. Copy your pas file to the directory that the LPS is scanning and your label should print. If it does not, refer to Chapter 1 for ideas on troubleshooting.
- 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.
- ActiveX Client control does not allow you to assign a drop directory to any directory that does not contain the subdir /status. The result is an error dialog. If you try to execute the Visual Basic application anyway, it shuts down, and you must 'end task' on the Visual Basic program, reload it, and revert to the last code save. For more information on Drop Directories and the ActiveX Client Control, see Knowledge Base Article #42715 on Loftware's Website.

Design Scenario and Distribution

Design Scenario

Suppose that you have written an application that you sell to end user accounts. These users would typically install your application on several client PCs, or, perhaps it is a server-based application.

You want to add the ability to print bar code labels from your application, but you do not want to have to pay for it unless your customer needs the barcoding feature. You need a solution that you can embed in your program that only has to be paid for if used.

You can do just that with the ActiveX Client Control. ActiveX Client Control is designed in such a way that you can embed it into your program and distribute it to your customers as part of your

application. When your customer decides that they want to use the barcoding capability of your application, you:

1. Sell them the Loftware product that contains the Loftware Print Server.
2. Install the LPS on the customer's network.
3. Activate the control in your program using a setup screen, registry setting, .ini file, etc.

*Note: You may install the LPS on the same PC as your application. Be aware, however, that you might pay a performance penalty if both applications are busy. **Note:** The ActiveX Client Control has a "thin footprint" and does not require the Loftware system to be installed on the same PC. It does, however, require that the LPS be installed somewhere on the network.*

Distributing the ActiveX Client Control with your Application

As mentioned above, the Client Control can be distributed free of charge for someone wishing to use the features that it provides. When building an install program for your application, include the required Loftware files with it. This way, your setup program takes care of installing the Loftware control and its associated files, thereby avoiding having to run two setup programs. The following is a list of the files that are needed to use the control. Please note that there are two versions of atl.dll, one for Windows 95(B)/98 and one for Windows NT/2000. Please do not mix them up! These files are found in the redistrib directory under the ActiveX Client directory on the Loftware CD. If your programming language has a setup wizard, it probably picks up the correct files by scanning our dependency file.

ATLClient.tlb -> winsys folder

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

atl.dll (for NT) v.3.00.8141 or greater ->winsys folder. This file needs to be registered.

atl.dll (for 95/98) v.2.00.7274 or greater->winsys folder. This file needs to be registered.

ActiveX Client Control Properties

Loftware's ActiveX Properties are listed below:

ClientIniPath Property	DropDirectory Property	Duplicates Property
FieldCount Property	FieldLength Property	FieldName Property
isRunning Property	JobName Property	LabelsPath Property
LayoutPath Property	Pages Property	PrinterAlias Property
PrinterCount Property	PrinterName Property	PrinterPort Property
PrinterNumber Property	PrinterPath Property	PasExt Property
Quantity Property	ServerCount Property	ServerNumber Property
ServerName Property	ServerAlias Property	TrimLeadingSpaces Property

ClientIniPath Property

Syntax

```
Object.ClientIniPath = path  
Type = string, read/write
```

Description

The ClientIniPath property allows the use of an "LLMWCInt.ini" file for the control to read the properties from the LPS. Upon setting this property, the file is parsed and an internal mapping of all servers and corresponding printers takes place. The llmwcInt.ini file needs to be created in a directory somewhere on your network. The "LabelsPath", "LayoutPath", "PrinterPath", "DropDirectory" as well as "ServerName" and "ServerAlias" properties for multiple servers are specified in the .ini file. If you are only using one Loftware Server (LPS), you need not use this property, and the other properties listed above can be set in code. The following syntax example shows an .ini file listing two LPS servers. Setting the "ClientIniPath" property to point to this file causes the control to be set up for multiple LPS servers. The ServerCount property is actually equal to 3 after setting the ClientIniPath to this .ini file. Server 0 is the default server that existed before the setting of ClientIniPath. Server 1 and 2 correspond to the ones listed in the .ini file.

The LLMWCInt.ini file can also be used to connect to multiple servers through socket connections. When used in this way, it is not necessary to specify the "Labels Path", "Layout Path", "Printer Path", "Drop Directory" or "ServerAlias" properties in the .ini file, only the "Server Name" and the "IPAddress" property are required. Any application using the ActiveX Control automatically logs in to all the servers listed in the LLMWCInt.ini upon execution of the program. Once you are connected to multiple servers, you can use the "SetServer" Method to move between servers.

Note: This property is what allows the Client Control to be aware of multiple Loftware Print Server Installations. See the section called "Performance Considerations" in Chapter 1 for information to help you to decide when you need to divide your printing load among multiple servers. Chapter 1 also has more information on the syntax requirements of the llmwcInt.ini file.

Syntax of llmwcInt.ini file:

```
[Receiving1]
Name=JANAA
Alias=Jana Computer
LabelsPath=c:\program files\Loftware Labeling\labels
LayoutPath=c:\program files\Loftware Labeling\Layouts
PrinterPath=c:\program files\Loftware Labeling
ScanPath=c:\program files\Loftware Labeling\wddrop
[DemoRoom]
Name=TRAINING
Alias=Training Room Server
LabelsPath=\\training\Loftware Labeling\labels
LayoutPath=\\training\Loftware Labeling\layouts
PrinterPath=\\training\Loftware Labeling
ScanPath=\\training\Loftware Labeling\wddrop
```

Example

'get a list of LPS servers and populate a list box with their names and 'aliases. This List could later be used to decide which server to set with the 'SetServer Method.

```
On Error GoTo Handler
frmFront.ClientX1.ClientIniPath = Trim(txtClientIniLocation.Text)
For i = 1 To frmFront.ClientX1.ServerCount - 1
    frmFront.ClientX1.SetServer (i)
    lstServers.AddItem frmFront.ClientX1.ServerName & " ALIAS " &
frmFront.ClientX1.ServerAlias
Next I
```

Syntax of llmwcInt.ini file using Socket Connections

```
[Shipping1]
Name=SHIPLINE1
Address=165.10.0.120
[Receiving1]
Name=RECEIVING1
Address=165.10.0.122
```

See also:

ServerCount	ServerAlias
ServerNumber	PrinterPath
ServerName	SetServer

DropDirectory Property

Syntax

```
dropDir = Object.DropDirectory  
Type = string, read/write if not using ClientIniPath, otherwise, read only
```

Description

The DropDirectory reflects the path to the Server's LPS scan path. This path is specified with a mapped drive or a UNC. Files are deposited in the scan directory for the Loftware Print Server to process when the PrintJob method is invoked. A good way to verify if your program is working properly is to shut down the LPS on the server and use explorer to view the folder specified with the DropDirectory property. A file is displayed after requesting a label with the PrintJob method with a .pas extension. View this file with an ASCII editor to verify that it is syntactically correct.

Note: If the LPS has multiple scan directories, it is up to your program to choose which directory to use for a given print job.

Example

```
` Set the DropDirectory for the Default server (0)  
ClientX1.DropDirectory = "\\XFILES\LOFTWARE\WDDrop"
```

See also

isRunning	PrinterPath	LayoutPath	LabelsPath
-----------	-------------	------------	------------

Duplicates Property

Syntax

```
Object.Duplicates = short  
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  
ClientX1.SetLabelName "Label1.lwl"  
ClientX1.SetData 0, "ABC-123"  
ClientX1.Quantity = 5  
ClientX1.Duplicates = 2  
ClientX1.PrintJob
```

See also

Quantity Property	Pages Property
-------------------	----------------

FieldCount Property

Syntax

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

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 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

```
'This example builds an SQL statement on the fly based on the field names  
'in the label. The database is then hit with a random key field and the  
'data for the fields is set. See the "Interface" section of Chapter 1 for 'more  
information on providing data for the general case.
```

Assumptions

```
'The field names in the label format are constrained to the field names in 'the  
database. (See Chapter 1.)
```

```
'Global myDatabase As Database defined in module1.
```

```
'Global myRecordset As Recordset defined in module1
```

```
'assume loftwarePath has been preset to the network location of the
```

```
'Loftware print server (LPS).
```

```
'GScanDirectory has been preset to the directory where the LPS is scanning.
```

```
Public Sub populateControlwithRandomData()
```

```
Dim sqlStatement As String, i As Integer, thisFieldName As String
```

```
'initialize the control to point to the LPS
```

```
ClientX1.LabelsPath = loftwarePath & "\labels"
```

```
ClientX1.LayoutPath = loftwarePath & "\layouts"
```

```
ClientX1.DropDirectory = gScanDirectory
```

```
ClientX1.PrinterPath = loftwarePath
```

```
'open the database
```

```
Set myDatabase = OpenDatabase(App.Path & "\sample.mdb")
```

```
'build SQL statement to grab data for this label
```

```
sqlStatement = "SELECT "
```

```
ClientX1.SetLabelName "mytest.LWL"
```

```
For i = 0 To ClientX1.FieldCount - 1
```

```
    sqlStatement = sqlStatement & "[" & ClientX1.FieldName(i) & "]" & ", "
```

```
Next i
```

```

'get rid of the last comma before the FROM clause and append key
sqlStatement = Left(sqlStatement, Len(sqlStatement) - 2) & " FROM Newwar WHERE
NAME1='" & cboRecordChoice.Text & "';"

'grab the record and populate the data
'only grab the fields we need for this label from the database
Err = 0
On Error Resume Next
Set myRecordset = myDatabase.OpenRecordset(sqlStatement)
If Err <> 0 Then
    MsgBox "SQL Error #" & Err & " SQL = " & sqlStatement,vbInformation, "SQL Error"
    Exit Sub
End If
' 'populate the label fields with the retrieved data
For i = 0 To ClientX1.FieldCount - 1
    thisFieldName = ClientX1.FieldName(i)
    ClientX1.SetData thisFieldName, myRecordset.Fields(thisFieldName)
Next i

myRecordset.Close
pickRandomRecord
myDatabase.Close
'print the label
ClientX1.PrintJob
End Sub

```

See also:

FieldName	FieldLength
-----------	-------------

FieldLength Property

Syntax

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

Description

The 'FieldCount', 'FieldName', 'FieldData', 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 pre-verifying data before actual printing by only allowing the amount of characters in the actual label field to be entered.

There is no default for this property.

Example

```
'By Field Index Number:  
Text1.MaxLength = ClientX1.FieldLength 0  
'By Field Name:  
Text1.MaxLength = ClientX1.FieldLength "PARTNUMBER"
```

See also:

FieldName	FieldCount
-----------	------------

FieldName Property

Syntax

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

Description

The 'FieldCount', 'FieldName', 'FieldData', and 'FieldLength' properties describe the array that 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

```
'This example builds an SQL statement on the fly based on the field names  
'in the label. The database is then hit with a random key field and the  
'data for the fields is set. See the "Interface" section of Chapter 1 for 'more  
information on providing data for the general case.
```

Assumptions

```
'The field names in the label format are constrained to the field names in 'the  
database. (See Chapter 1)
```

```
'Global myDatabase As Database defined in module1.
```

```
'Global myRecordset As Recordset defined in module1
```

```
'assume loftwarePath has been preset to the network location of the
```

```
'Loftware print server (LPS).
```

```
'GScanDirectory has been preset to the directory where the LPS is
```

```
'scanning.
```



```

Public Sub populateControlwithRandomData()
Dim sqlStatement As String, i As Integer, thisFieldName As String
'initialize the control to point to the LPS
ClientX1.LabelsPath = loftwarePath & "\labels"
ClientX1.LayoutPath = loftwarePath & "\layouts"
ClientX1.DropDirectory = gScanDirectory
ClientX1.PrinterPath = loftwarePath
'open the database
Set myDatabase = OpenDatabase(App.Path & "\sample.mdb")
'build SQL statement to grab data for this label
sqlStatement = "SELECT "
ClientX1.SetLabelName "mytest.LWL"
For i = 0 To ClientX1.FieldCount - 1
    sqlStatement = sqlStatement & "[" & ClientX1.FieldName(i) & "]" & ", "
Next i
'get rid of the last comma before the FROM clause and append key
sqlStatement = Left(sqlStatement, Len(sqlStatement) - 2) & " FROM Newwar WHERE
NAME1='" & cboRecordChoice.Text & "';"
'grab the record and populate the data
'only grab the fields we need for this label from the database
Err = 0
On Error Resume Next
Set myRecordset = myDatabase.OpenRecordset(sqlStatement)
If Err <> 0 Then
    MsgBox "SQL Error #" & Err & " SQL = " & sqlStatement, vbInformation, "SQL Error"
    Exit Sub
End If

'populate the label fields with the retrieved data
For i = 0 To ClientX1.FieldCount - 1
    thisFieldName = ClientX1.FieldName(i)
    ClientX1.SetData thisFieldName, myRecordset.Fields(thisFieldName)
Next i
myRecordset.Close
pickRandomRecord
myDatabase.Close
'print the label
ClientX1.PrintJob
End Sub

```

See also:

FieldCount	FieldLength
------------	-------------

isRunning Property

Syntax

```
Object.isRunning = boolean  
Type = boolean
```

Description

The isRunning property checks to see if the currently selected Loftware Print Server is scanning. If the LPS is not found, or is not scanning, error # 25518 is thrown in the ErrorEvent. Make sure that you handle this error in the ErrorEvent, if you do NOT, your program may crash!

Example

```
On Error Resume Next  
Err = 0  
If ClientX1.isRunning = False Then  
    MsgBox("LPS is not running!")  
End If  
If Err Then  
    If (Err.Number - &H80000000 = 25518) then  
        MsgBox "LPS is not reachable"  
    End If  
End If
```

See also:

DropDirectory	PrinterPath	LabelsPath
-------------------------------	-----------------------------	----------------------------

JobName Property

Syntax

```
jobName = 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 + "_" + YYYYMMDDHHNNSS + serial number per second

If you wish to specify your own JobName, it is up to the creator to ensure its uniqueness across time as well as space. 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 = ClientX1.JobName
```

See also:

PrintJob	AppendJob	PrinterNumber
--------------------------	---------------------------	-------------------------------

LabelsPath Property

Syntax

```
LabelsPath = Object.LabelsPath
Type = string, read/write if not using ClientIniPath, otherwise, read only
```

Description

LabelsPath is a read only property for servers listed in the LLMWCInt.ini file and writeable for the default server (Server 0). The LabelsPath reflects the UNC or mapped drive path to the labels directory that the server is using. When LabelsPath is used, the LabelsPath is prepended to label files in the call to SetLabelName if the path is not present.

If you receive "Format not found" errors, chances are the LabelsPath is incorrect or not mapped properly from the client. Error number 25513 may be thrown if the path does not exist. Handle this error in the ErrorEvent to prevent your program from crashing!

Example

```
` Set the LabelsPath for the Default server (0)
ClientX1.LabelsPath = "\\XFILES\SOFTWARE\Labels"
```

See also:

LayoutPath	PrinterPath
----------------------------	-----------------------------

LayoutPath Property

Syntax

```
layoutPath = Object.LayoutPath
Type = string, read/write if not using ClientIniPath, otherwise, read only
```

Description

The LayoutPath reflects the UNC or mapped drive path to the Server's Layout directory.

LayoutPath is currently not used by the control and is reflected due to its existence in the LLMWCInt.ini file.

Example

```
` Set the LayoutPath for the Default server (0)
ClientX1.LayoutPath = "\\XFILES\LOFTWARE\Layouts"
```

See also:

LabelsPath	PrinterPath
------------	-------------

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 LLM-WIN manual. You need not use this property if your label does not use a "multi-up" layout.

Example

```
'print 2 identical pages, 10 labels/page with 2 labels for each serial #
ClientX1.SetLabelName "Label1.lwl"
ClientX1.SetData 0, "ABC-123"
ClientX1.Quantity = 5
ClientX1.Duplicates = 2
ClientX1.Pages = 2
ClientX1.PrintJob
```

See also:

Quantity	Duplicates	PrinterAlias
----------	------------	--------------

PrinterAlias Property

Syntax

```
printerAlias = Object.PrinterAlias
Type = string, read only
```

Description

PrinterAlias is used along with PrinterName, PrinterCount, and PrinterPort properties to allow you to prompt your users for the target-configured printer to whom 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 device configuration menu of label design mode (same dialog box where you specify a port/spool/IP address).

Note: The PrinterPath or the ClientIniPath property MUST be set in order to obtain the printer information. The Printer properties only reflect the printers that have been previously configured at the LPS. They are Read Only!

Example

'this example populates a list box with printer aliases. If the
'printer alias does not exist, it uses the default printer name.

```
Public Sub populatePrinterList()  
Dim i As Integer  
If ClientX1.PrinterCount = 0 Then  
    MsgBox "You have not configured any printers in Loftware yet."  
    Exit Sub  
End If  
For i = 1 To ClientX1.PrinterCount  
    If ClientX1.PrinterAlias(i) = "" Then  
        frmFront.lstPrinters.AddItem ClientX1.printerName(i)  
    Else  
        frmFront.lstPrinters.AddItem ClientX1.PrinterAlias(i)  
    End If  
Next i  
End Sub
```

See also:

PrinterName	PrinterCount
-------------	--------------

PrinterCount Property

Syntax

```
numConfiguredPrinters = 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 Loftware label design mode as CLIENT DEFINED.

Note: Either the PrinterPath or the ClientIniPath property MUST be set in order to obtain the printer information. The Printer properties only reflect the printers that have been previously configured at the LPS. They are Read Only!

Example

```
'this example populates a list box with printer aliases. If the printer
'alias does not exist, it uses the default printer name.
Public Sub populatePrinterList()
Dim i As Integer
If ClientX1.PrinterCount = 0 Then
    MsgBox "You have not configured any printers in Loftware Yet."
    Exit Sub
End If
For i = 1 To ClientX1.PrinterCount
    If ClientX1.PrinterAlias(i) = "" Then
        frmFront.lstPrinters.AddItem ClientX1.printerName(i)
    Else
        frmFront.lstPrinters.AddItem ClientX1.PrinterAlias(i)
    End If
Next i
End Sub
```

See also:

PrinterName	PrinterAlias
-----------------------------	------------------------------

PrinterName Property

Syntax

```
printerName = Object.PrinterName
Type = string, read only
```

Description

PrinterName is used along with PrinterCount, PrinterPort, and PrinterAlias properties to allow you to be able 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.

Note: The PrinterPath or the ClientIniPath property MUST be set in order to obtain the printer information. The Printer Properties only reflect the printers that have been previously configured at the LPS. They are Read Only!

Example

```
'this example populates a list box with printer aliases. If the printer
'alias does not exist, it uses the default printer name.
Public Sub populatePrinterList()
Dim i As Integer
If ClientX1.PrinterCount = 0 Then
    MsgBox "You have not configured any printers in Loftware Yet."
    Exit Sub
End If
```

```

For i = 1 To ClientX1.PrinterCount
    If ClientX1.PrinterAlias(i) = "" Then
        frmFront.lstPrinters.AddItem ClientX1.printerName(i)
    Else
        frmFront.lstPrinters.AddItem ClientX1.PrinterAlias(i)
    End If
Next i
End Sub

```

See also:

PrinterAlias

PrinterPort Property

Syntax

```

printerPort = Object.PrinterPort
Type = string, read only

```

Description

PrinterPort is used along with PrinterName, PrinterCount, and PrinterAlias properties to allow you to be able 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 is set up in the Loftware label design mode.

Note: Either the PrinterPath or the ClientIniPath property MUST be set in order to obtain the printer information. The Printer Properties only reflect the printers that have been previously configured at the LPS. (They are Read Only!)

Example

```

`this example populates a list with configured printer ports
Public Sub populatePorts()
Dim i As Integer, gotOne As Boolean
For i = 1 To ClientX1.PrinterCount
    If ClientX1.printerName(i) <> "Not Configured" Then
        frmFront.lstPorts.AddItem ClientX1.PrinterPort(i)
    End If
Next i
End Sub

```

See also:

PrinterPath	ClientIniPath	PrinterName
-------------	---------------	-------------

PrinterNumber Property

Syntax

```
Object.PrinterNumber = short  
Type = string, read/write
```

Description

The PrinterNumber property is a read/write property that sets which label formats to print to which configured LPS printer. 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

```
'Printer 1 in Loftware might be defined as: Loftware Intermec 3440 on COM2:  
'Printer 2 in Loftware might be defined as: HP LaserJet 4M on \\Server\Print\HP  
'In this example, to print to the Intermec printer, set the PrinterNumber  
'property to 1.  
ClientX1.SetLabelName "Label1.lw1"  
ClientX1.SetData 0, "ABC-123"  
ClientX1.PrinterNumber = 1  
ClientX1.PrintJob  
'To print to the HP printer, set the PrinterNumber property to 2:  
ClientX1.SetLabelName "Label2.lw1"  
ClientX1.SetData 0, "ABC-123"  
ClientX1.PrinterNumber = 2  
ClientX1.PrintJob
```

See also:

SetLabelName	SetData	PrintJob
--------------	---------	----------

PrinterPath Property

Syntax

```
printerPath = Object.PrinterPath  
Type = string, read/write if not using ClientIniPath, otherwise, read only
```

Description

The PrinterPath reflects the path to the LPS install directory or the path that contains the printr32.ini file. The LPS path on the server path would be "drive:\Program Files\Loftware Labeling" by default.

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

Example

```
` Set the PrinterPath for the Default server (0)
ClientX1.PrinterPath = "\\XFILES\LOFTWARE"
```

See also:

LayoutPath	LabelsPath	ClientIniPath
------------	------------	---------------

PasExt Property

Syntax

```
pasExt = Object.PrinterPath
Type = string, read/write if not using ClientIniPath, otherwise, read only
```

Description

The PasExt Property reflects the extension that the LPS is expecting pass files to use. If files are not processed in the LPS scan directory, it may be because the server is looking for files of a different extension. The default pass file extension is ".pas", which is the default for the server as well. The "." should not be specified when setting this property.

Example

```
` Set the PasExt for the Default server (0)
` This is for non-standard extensions only
ClientX1.PasExt = "pvv"
```

See also:

ServerCount	ServerNumber	ServerName	DropDirectory
ClientIniPath	SetServer	IsRunning	

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  
ClientX1.SetLabelName "Label1.lw1"  
ClientX1.SetData 0, "ABC-123"  
ClientX1.Quantity = 5  
ClientX1.Duplicates = 2  
ClientX1.PrintJob
```

See also:

Duplicates

Pages

ServerCount Property

Syntax

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

Description

ServerCount is used along with ServerNumber, ServerName, ServerAlias, and ClientIniPath properties. ServerCount only reflects a value greater than 1 (one) if a valid ClientIniPath & configuration file (LLMWCInt.ini) with multiple servers listed is set. The default for this property is 1 (the Default Server). This property is useful for iterating through a list of servers to get the names and aliases for display to the user. If you are not using the ClientIniPath property with multiple servers specified in the .ini file, ServerCount has no use.

Example

```
'this example sets the ClientIniPath property to a .ini file name. It then  
'populates a list box control with the server names and aliases found in  
'the ini file. The iteration is controlled with the ServerCount property.
```

```
Dim i As Integer  
On Error GoTo Handler  
frmFront.ClientX1.ClientIniPath = Trim(txtClientIniLocation.Text)  
For i = 1 To frmFront.ClientX1.ServerCount - 1  
    frmFront.ClientX1.SetServer (i)  
    lstServers.AddItem frmFront.ClientX1.ServerName & " ALIAS " &  
    frmFront.ClientX1.ServerAlias  
Next i
```

See also:

ServerNumber	ServerName	ServerAlias
PrinterPath	SetServer	ClientIniPath

ServerNumber Property

Syntax

```
serverNumber = Object.ServerNumber  
Type = short, read only
```

Description

ServerNumber is used along with ServerCount, ServerName, and ServerAlias properties. ServerNumber reflects the index of the internal Server array. The default for this property is 0. This property enables you to retrieve, in code, the number of the currently selected server that was selected with the SetServer Method. If you are not using the ClientIniPath property with multiple servers specified in the .ini file, ServerNumber has no use.

Example

```
'this example gets the currently selected server and sets the system to the  
'next one. If the next one goes beyond server count, 0 is used  
'(the default server)
```

```
Dim I as integer  
serverNumber = ClientX1.ServerNumber  
' Now let us get the next server  
serverNumber = serverNumber + 1  
If (serverNumber = ClientX1.ServerCount) then  
    serverNumber = 0  
end if  
ClientX1.SetServer(serverNumber)
```

See also:

ServerCount	ServerName	ServerAlias
PrinterPath	SetServer	

ServerName Property

Syntax

```
serverName = Object.ServerName  
Type = string, read only
```

Description

ServerName is used along with ServerCount, ServerNumber, and ServerAlias properties. ServerName reflects the name of the currently selected Server. If you are not using the ClientIniPath property with multiple servers specified in the .ini file, ServerName has no use.

Example

'this example sets the ClientIniPath property to a .ini file name. It then 'populates a list box control with the server names and aliases found in 'the ini file. The iteration is controlled with the ServerCount property.

```
Dim i As Integer
On Error GoTo Handler
frmFront.ClientX1.ClientIniPath = Trim(txtClientIniLocation.Text)
For i = 1 To frmFront.ClientX1.ServerCount - 1
    frmFront.ClientX1.SetServer (i)
    lstServers.AddItem frmFront.ClientX1.ServerName & " ALIAS " &
    frmFront.ClientX1.ServerAlias
Next i
```

See also:

ServerCount	ServerNumber	ServerAlias
PrinterPath	SetServer	ClientIniPath

ServerAlias Property

Syntax

```
serverAlias = Object.ServerAlias
Type = string, read only
```

Description

ServerAlias is used along with ServerCount, ServerNumber, and ServerName properties. ServerAlias reflects the alias name of the currently selected Server. If you are not using the ClientIniPath property with multiple servers specified in the .ini file, ServerAlias has no use.

Example

'this example sets the ClientIniPath property to a .ini file name. It then 'populates a list box control with the server names and aliases found in 'the ini file. The iteration is controlled with the ServerCount property.

```
Dim i As Integer
On Error GoTo Handler
frmFront.ClientX1.ClientIniPath = Trim(txtClientIniLocation.Text)
For i = 1 To frmFront.ClientX1.ServerCount - 1
    frmFront.ClientX1.SetServer (i)
    lstServers.AddItem frmFront.ClientX1.ServerName & " ALIAS " &
    frmFront.ClientX1.ServerAlias
Next i
```

See also:

ServerCount	ServerNumber	ServerName
PrinterPath	SetServer	ClientIniPath

TrimLeadingSpaces Property

Syntax

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

Description

The TrimLeadingSpaces property trims the leading spaces from data before it presented to the label. 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  
ClientX1.SetLabelName "Label1.lwl"  
ClientX1.TrimLeadingSpaces=TRUE  
ClientX1.SetData 0, " ABC-123"  
ClientX1.PrintJob
```

ActiveX Methods

Loftware's ActiveX Methods are:

AppendJobMethod	ClearData Method
Login Method	PrintJob Method
ResetJob Method	SetData Method
SetLabelName Method	SetServer Method

AppendJob Method

Syntax

```
Object.AppendJob
```

Description

AppendJob is used to queue up the current label that has been designated via the SetData 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. The resulting .pas file that is placed in the DropDirectory contains a "stacked" list of all requested labels.

Note: If the PrinterNumber property is not set before invoking this method, the default printer (Printer 1) is used.

Example

```
'Queue up several labels and then print

For j = 1 To numberOfLabels
  ClientX1.ClearData
  populateControlwithRandomData
  ClientX1.Quantity = MaskQuantity.Text
  ClientX1.Duplicates = maskDuplicates.Text
  ClientX1.printerNumber = printerNumber
  ClientX1.AppendJob
Next j
'the batch is now complete, submit the job to LPS
ClientX1.PrintJob
```

See also:

PrintJob	SetData	SetLabelName
--------------------------	-------------------------	------------------------------

ClearData Method

Syntax

```
Object.ClearData
```

Description

ClearData 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

```
'Queue up several labels and then print
For j = 1 To numberOfLabels
  ClientX1.ClearData
  populateControlwithRandomData
  ClientX1.Quantity = MaskQuantity.Text
  ClientX1.Duplicates = maskDuplicates.Text
  ClientX1.printerNumber = printerNumber
  ClientX1.AppendJob
Next j
` The Batch is complete, now send it
ClientX1.PrintJob
```

See also:

[ResetJob](#)

Login Method

Syntax

```
Object.Login (IPAddress as string, ServerName as string)
```

Description

The *Login* method is the fastest and easiest way to login to a single LPS through a socket connection. When called, the ActiveX Client Control issues a login request to the LPS. The LPS must be online and started. The LPS then responds and opens a communication socket between itself and the application using the ActiveX Client Control. Once connected, information is sent back and forth between the client and the server, including any properties or methods used during the current session.

Example

```
`use the Login method connect to connect to a single LPS
`The following issues a Login Request on an LPS running on a server named
`SHIPLINE1 with an IP Address of 165.10.0.120
ClientX1.Login "165.10.0.120", "SHIPLINE1"
```

PrintJob Method

Syntax

```
Object.PrintJob
```

Description

PrintJob is used to submit the current job to the LPS. The current data is supplied via the SetData method calls, as well as Quantity, Duplicates, Pages, and/or PrinterNumber properties. A .pas file containing the request is written to the DropDirectory after executing this method. The LPS subsequently picks it up and prints it.

Example

```
`initialize the control to point to the LPS
`assume loftwarePath has been preset to the network location of
`the Loftware print server (LPS). GScanDirectory has been preset
`to the directory where the LPS is scanning.
ClientX1.LabelsPath = loftwarePath & "\\labels"
ClientX1.LayoutPath = loftwarePath & "\\layouts"
ClientX1.DropDirectory = gScanDirectory
ClientX1.PrinterPath = loftwarePath
ClientX1.SetLabelName "mytest.LWL"
`populate the data fields with data typed into a text input control array
For i = 0 to ClientX1.FieldCount - 1
    ClientX1.SetData(i) txtfield(i)
Next i
ClientX1.PrintJob
```

See also:

SetLabelName	SetData	AppendJob
--------------	---------	-----------

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

```
` Want to send x number of labels in one job to the printer
dim inner as integer
dim outer as integer
For outer = 0 to x
    'Ensure all the data for the current label is cleared
    ClientX1.ClearData
    for inner = 0 to ClientX1.FieldCount -1
        ClientX1.SetData inner, "Some Data"
    Next inner
    'Append this label to the batch
    ClientX1.AppendJob
Next outer
` Did the user click the cancel button?
If cmdCancel.Enabled = FALSE then
    ` Yep, lets u delete all this data
    ClientX1.ResetJob
    Exit function ` And leave
Else
    `The Batch is complete, now send it
    ClientX1.PrintJob
End If
```

See also:

ClearData

SetData Method

Syntax

Object.SetData FieldName or index as Variant, Data as String

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'.

Example 1

```
'Sending data to the array by using the field's index number:
'setting label data by iterating through field count
ClientX1.SetLabelName "mytest.LWL"
'populate the data fields with data typed into a text input control array
For i = 0 to ClientX1.FieldCount - 1
    ClientX1.SetData(i) txtfield(i)
Next i
ClientX1.PrintJob
```

Example 2

'This example builds an SQL statement on the fly based on the field names in the label. The database is then hit with a random key field and the data for the fields is set. See the "Interface" section of Chapter 1 for more information on providing data for the general case.

Assumptions

'The field names in the label format are constrained to the field names in the database. (See Chapter 1)
'Global myDatabase As Database defined in module1.
'Global myRecordset As Recordset defined in module1
'assume loftwarePath has been preset to the network location of the
'Loftware Print Server (LPS).
'GScanDirectory has been preset to the directory where the LPS is scanning.

```
Public Sub populateControlwithRandomData()
Dim sqlStatement As String, i As Integer, thisFieldName As String
'initialize the control to point to the LPS
ClientX1.LabelsPath = loftwarePath & "\labels"
ClientX1.LayoutPath = loftwarePath & "\layouts"
ClientX1.DropDirectory = gScanDirectory
ClientX1.PrinterPath = loftwarePath
'open the database
Set myDatabase = OpenDatabase(App.Path & "\sample.mdb")
'build SQL statement to grab data for this label
sqlStatement = "SELECT "
ClientX1.SetLabelName "mytest.LWL"
For i = 0 To ClientX1.FieldCount - 1
    sqlStatement = sqlStatement & "[" & ClientX1.FieldName(i) & "]" & ", "
Next i
'get rid of the last comma before the FROM clause and append key
sqlStatement = Left(sqlStatement, Len(sqlStatement) - 2) & " FROM Newwar WHERE
```

```

NAME1="" & cboRecordChoice.Text & "';"

'grab the record and populate the data
'only grab the fields we need for this label from the database
Err = 0
On Error Resume Next
Set myRecordset = myDatabase.OpenRecordset(sqlStatement)
If Err <> 0 Then
    MsgBox "SQL Error #" & Err & " SQL = " & sqlStatement, vbInformation, "SQL Error"
    Exit Sub
End If
'populate the label fields with the retrieved data
For i = 0 To ClientX1.FieldCount - 1
    thisFieldName = ClientX1.FieldName(i)
    ClientX1.SetData thisFieldName, myRecordset.Fields(thisFieldName)
Next i
myRecordset.Close
pickRandomRecord
myDatabase.Close
'print the label
ClientX1.PrintJob
End Sub

```

Example3:

```

'setting data by iterating through field count instead of using field name
'this example sets every field to "sample data"
ClientX1.SetLabelName "ibm.LWL"
For I = 0 to ClientX1.FieldCount - 1
    ClientX1.SetData I, "sample data"
Next I
ClientX1.PrintJob

```

See also:

SetLabelName	PrintJob	LabelsPath
LayoutPath	DropDirectory	PrinterPath

SetLabelName Method

Syntax

```
Object.SetLabelName (newLabelName as String)
```

Description

The SetLabelName method 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 iterated by index number or field name. Obviously, 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.

The SetLabelName Method does the following: Creates an Array to hold label field names, maximum field lengths, and data.

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

```
'assume loftwarePath has been preset to the network location of the
'Loftware print server (LPS). GScanDirectory has been preset to the
'directory where the LPS is scanning.
ClientX1.LabelsPath = loftwarePath & "\labels"
ClientX1.LayoutPath = loftwarePath & "\layouts"
ClientX1.DropDirectory = gScanDirectory
ClientX1.PrinterPath = loftwarePath
ClientX1.SetLabelName "AIAG.LWL"
ClientX1.SetData "PARTNUMBER", "A100"
ClientX1.PrintJob
```

See also:

PrintJob	SetData	LabelsPath
LayoutPath	DropDirectory	PrinterPath

SetServer Method

Syntax

Object.SetServer ServerName or index as Variant

Description

SetServer is only useful if you are using the ClientIniPath property to specify multiple Loftware servers in the llmwclnt.ini file. After pointing the ClientIniPath property to a .ini file that contains references to multiple Loftware servers, you may use the SetServer method to tell the control which server to use. The Labels, Layouts, and Printers paths automatically inherit the values specified in the corresponding entry in the .ini file. If you need to disable a previous use of the ClientIniPath property, "SetServer 0" causes the default server to be selected. That uses the "LabelsPath", "LayoutPath", "DropDirectory", and "PrinterPath" properties manually set with the control instead of the ones that are specified in the .ini file. Therefore, server numbers specified in the .ini file start at 1.

Example

```
'this example sets the ClientIniPath property to a .ini file name. It then
'populates a list box control with the server names and alias's found in
```

'the ini file. The iteration is controlled with the ServerCount property, with 'the selected server being controlled with the SetServer Method. Notice the 'default server (0) is not included in the loop.

```
Dim i As Integer
On Error GoTo Handler
frmFront.ClientX1.ClientIniPath = Trim(txtClientIniLocation.Text)
For i = 1 To frmFront.ClientX1.ServerCount - 1
    frmFront.ClientX1.SetServer (i)
    lstServers.AddItem frmFront.ClientX1.ServerName & " ALIAS " &
    frmFront.ClientX1.ServerAlias
Next i
```

See also:

ServerCount	ServerName	ServerAlias	PrinterPath	ClientIniPath
-------------	------------	-------------	-------------	---------------

ActiveX Events

Loftware's ActiveX Events are described in this section.

ErrorEventEvent	WarningEventEvent
OtherEventEvent	InfoEventEvent

ErrorEvent Event

Syntax

errorID as Long, errorString as String

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 warningevents 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 ClientX1_ErrorEvent(ByVal errorID As Long, ByVal errorString
As String)
Dim nResult, szMsg
szMsg = "Error or Failover event -" & vbNewLine
```

```

szMsg = szMsg & "Loftware ActiveX Error Event #" & errorID & " has
occurred. Error String = " & Trim(errorString) & ". Consult the
documentation for the Loftware ActiveX Client control for more
information." & vbNewLine & vbNewLine
szMsg = szMsg & "Cluster Failover -" & vbNewLine
szMsg = szMsg & "If your Loftware Print Server is running in a clustered
environment, you may be experiencing a failover situation. Wait 30
seconds and try your program again."
nResult = MsgBox(szMsg, vbCritical, "Loftware - Critical Error or
Failover in Progress")
End
End Sub

```

ErrorEvent Ids and Messages (Also includes error returned from the control)

ID	Message	Explanation
25500	Name not found!	Name entry in llmwcint.ini file not found.
25501	PrinterPath not found!	PnnterPath entry in llmwcInt.ini file not found
25502	Unable to locate path to Printr32.ini file!	While attempting to obtain the printer information, the control could not locate the path for the printer information.
25503	Critical Failure of job # jobNum (Not currently used in the control)	A critical failure of a job has been reported
25504	Error on printer printrnumber	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 printr32.ini file
25506	Printer Port: Not Found	The printer port was not located in the printr32.ini file.
25507	Printer ID: Not Found	The printer id was not located in the printr32.ini file.
25508	No servers located. Application cannot print labels!	There were no server entries (sections) located in the llmwcInt.ini file.
25510	Redirected Ini File does not exist!	The redirected ini file was not located.
25512	Failed to obtain the Computer Name!	The computer name is required to generate the unique jobname as well as unique filenames.
25513	Path does not exist!	Tested path does not exist!
25514	Write Access Denied!	Write access (For the DropDirectory) has been denied during the test.
25515	Read Access Denied for File: file	Cannot read file in Label, Layout, or PrinterPath.
25516	File does not exist!	Tested file does not exist (printr32.ini).
25517	Read Access Denied!	Read access for a given directory is denied (LabelsPath, LayoutPath, or PrinterPath).
25518	Sever name is stopped	Recorded during isRunning call if server is not running. (Yet accessible)
25519	Server name is unreachable!	Recorded during isRunning call if the DropDirectory is not set, or is incorrect.
26500	Unable to open file.	Unable to open the label file (unlikely)

ID	Message	Explanation
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 with version 3.5.2.58 of llmwdn32.exe or greater.
26504	Label not set	An attempt to access a field or print a job with having called 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'.
26507	Printerpath is not set	An attempt was made to access printer information Printlob without first setting the PrinterPath or ClientiniPath, or the path was determined invalid.
26508	DropDirectory is not set.	An attempt to call Printjob without setting the DropDirectory or ClientiniPath. Or the path was determined invalid
26509	Not a Valid Printer Number	A negative printer number was passed
26510	Printeralias 'alias' not found	An invalid printer alias was passed to PrinterPort.
26511	Error creating pass file 'file'.	There was an error while attempting to create a pass file in the DropDirectory.
26512	ClientIni file 'file' not found	The llmwcInt.ini file was not located in the specified path (ClientIniPath).
26513	No sections located in ini file.	There were no server sections located in the llmwcInt.ini file.
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 fldname.
26516	Invalid Server Name	SetServer was passed a ServerName that does not exist.
26517	Invalid Server Number	SetServer was passed a server number or index that does not exist.
25520		Printer Error ...job should print when error is cleared.
25261		All requests failed (Critical)
25262		Stacked job partially printed
27000		Ignore

See also:

WarningEvent	OtherEvent	InfoEvent
--------------	------------	-----------

InfoEvent Event

Syntax

```
infoID as Long, infoString as String
```

Description

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 infoevents 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 ClientX1_InfoEvent(ByVal infoID As Long, ByVal infoString As String)  
    ' Log these messages to an info multi-line textbox  
    lstInfoList.AddItem infoString  
End sub
```

InfoEvent IDs and Messages

ID	Message	Explanation
25000	LabelsPath: path	Displays the LabelsPath.
25001	LayoutPath: path	Displays the Layout Path.
25002	PrinterPath: path	Displays the PrinterPath.
25003	ScanPath: path	Displays the DropDirectory.
25004	StatusPath: path	Displays the internal path used for status.
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 (Not used)
25013	Printer Alias: alias	Displays the PrinterAlias
25016	Looking for ini File: file	Checking for the existence of the LLMWCInt.ini file in the ClientIniPath.
25017	Using Redirected ini File: file	Detected the use of a redirected LLMWCInt.ini file and attempts to obtain config information from the listed file.
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
25029		Job Printed

WarningEvent Event

Syntax

warningID as Long, warningString as String

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 ClientX1_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.
25251	LabelsPath not found	There was no entry for LabelsPath in the llmwcInt.ini file, the default is the "Labels" sub-directory off the PrinterPath.
25252	LayoutPath not found	There was no entry for LayoutPath in the llmwcInt.ini file, the default is the "Layouts" sub-directory off the PrinterPath.
25253	ScanPath not found	There was no entry for ScanPath in the llmwcInt.ini file, the default is the "WDDrop" sub-directory off the PrinterPath
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).

See also:

ErrorEvent	OtherEvent	InfoEvent
------------	------------	-----------

OtherEvent Event

Syntax

```
otherID as Long, otherString as String
```

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 ClientX1_OtherEvent(ByVal otherID As Long, ByVal otherString As String)
` Log these messages to an info multi-line textbox
txtOther.Text = otherString
` No ID tests as there are no known other Ids. . .
End sub
```

See also:

WarningEvent	ErrorEvent	InfoEvent
--------------	------------	-----------

ActiveX Client Control Reference Table

Name	Read	Write	Type	Default	Comment
<i>AppendJob</i>	N/A	N/A	Method	N/A	Batches label requests together.
<i>Clear Data</i>	N/A	N/A	Method	N/A	Clears all the data members for the current label
<i>ClientIniPath</i>	X	X	Property	N/A	Allows use of LLMWCInt.ini file for the control to read the properties from the LPS, or to connect to multiple .servers through socket connections.
<i>DropDirectory</i>	X	X*	Property	N/A	Reflects the path to the server's LPS scan path.
<i>Duplicates</i>	X	X	Property	1	Sets the amount of duplicate labels to print.
<i>ErrorEvent</i>	N/A	N/A	Event	N/A	Passes error messages back to the container.
<i>Field Count</i>	X	X	Property	N/A	Displays how many variable fields there are in the current label format
<i>FieldLength</i>	X	N/A	Property	N/A	Displays the length of a specified field in the current label format
<i>FieldName</i>	X	X	Property	N/A	Displays the name of a specific field in the current label format.
<i>InfoEvent</i>	N/A	N/A	Event	N/A	Passes informational messages back to the container.
<i>isRunning</i>	N/A	N/A	Property	N/A	Checks to see if the selected LPS is started.
<i>JobName</i>	X	X	Property	Contains computer name	Reflects a unique identifier for the current job.
<i>LabelsPath</i>	X	X*	Property	N/A	Reflects the UNC or mapped drive path to the labels directory used by the LPS.
<i>LayoutPath</i>	X	X*	Property	N/A	Reflects the UNC or mapped drive path to the LPS's Layout Directory.
<i>Login</i>	N/A	N/A	Method	N/A	Used to log in to a single LPS through a socket connection.
<i>OtherEvent</i>	N/A	N/A	Event	N/A	Contains information that does not fit into the info, warning or error categories
<i>Pages</i>	X	X	Property	1	Sets how many pages of labels are printed.
<i>PrinterAlias</i>	X		Property	N/A	The descriptive name for an assigned printer.

Name	Read	Write	Type	Default	Comment
<i>Printer Count</i>	X		Property	N/A	The number of configured printers.
<i>PrintJob Method</i>	N/A	N/A	Method	N/A	Used to submit the current job to the LPS.
<i>PrinterName</i>	X		Property	N/A	Name of configured printer.
<i>PrinterPort</i>	X		Property	N/A	List of configured printer ports
<i>PrinterNumber</i>	X	X	Property	1	Corresponds to configured Printers
<i>PrinterPath</i>	X	X*	Property	N/A	Reflects the path to the LPS install directory.
<i>PasExt</i>	X	X*	Property	.pas	Reflects the extension that the LPS is expecting pass files to use.
<i>Quantity</i>	X	X	Property	1	Sets the amount of labels to print.
<i>ResetJob</i>	N/A	N/A	Method	N/A	Clears the entire job, including appended labels.
<i>ServerCount</i>	X		Property	N/A	Iterates through a list of servers to display names and aliases to the user.
<i>ServerNumber</i>	X		Property	N/A	Number of currently selected server.
<i>ServerName</i>	X		Property	N/A	The name of the currently selected server.
<i>ServerAlias</i>	X		Property	N/A	The alias name of the currently selected server.
<i>SetData</i>	N/A	N/A	Method	N/A	Populates the field by index number or field name.
<i>SetLabelName</i>	N/A	N/A	Method	N/A	Retrieves field information for the specified label format.
<i>SetServer</i>	N/A	N/A	Method	N/A	Tells the control which server to use.
<i>TrimLeading Spaces</i>			Property	0	Trims the leading spaces from the data before it is presented to the label.
<i>WarningEvent</i>			Event	N/A	Passes warning messages back to the container during various operations.

*Only if not using ClientIniPath



Chapter 6 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.

Found in this chapter is 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 Loftware Print Server functions with their own applications. In the previous ActiveX chapters, Loftware 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.

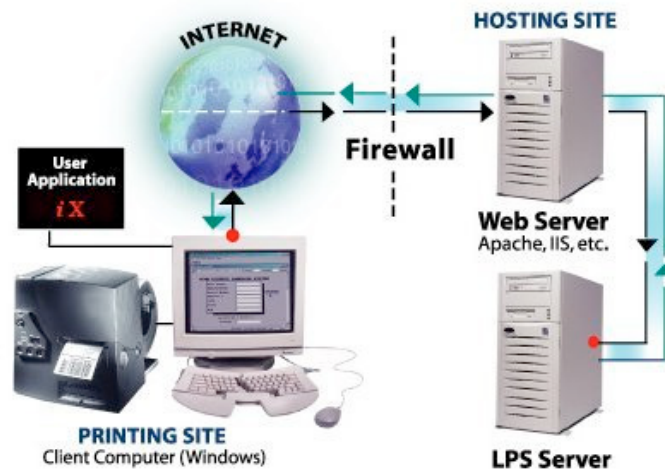


Figure 6-A: The Concept of Internet ActiveX

This chapter documents the Loftware Internet ActiveX Control. Like the WebClient, and our 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?

1. **The Software Print Server (LPS)** - Refer to Chapter 1 for instructions on how to do this.
 - Please note that the LPS can be installed on any NT/2000/XP computer anywhere on your LAN or WAN.
2. **The Web Server** – refer to information found in Chapter 3.
3. **Servlet Engine** - refer to information found in Chapter 3.
4. **LPS Web Servlet** - refer to instructions found in Chapter 3.
5. **Client-defined Printers** – refer to information and instructions found in Chapter 2 of the LLM User's Guide and in Chapter 2 of this User's Guide.

Hint: *Install and run the WebClient from a Client PC to ensure that the LPSi installation is successful before proceeding.*

Installing the Internet ActiveX Control

Install the Internet ActiveX Control to the Client PC one of the 3 following ways:

1. A "Full" install from the CD adds the *web x dll*.
2. A Client install from the *LPS Client Install* Folder on the CD.
3. 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

1. Load the sample program, called **Internet ActiveX Sample 1**, learn and study it. (This saves you time in the long run.)
 - Sample programs are installed with the LPS Premier Edition. They are present in a subdirectory called sample programs.
 - The sample program can be run from the PC where Software is installed or a client PC. To run the program from the PC where Software is installed you do not have to perform a Web Client Install. If running the program from a client PC, perform the **LPS WebClient Install** that is found in the **LPS Client Install** folder on the Software CD.
 - Configure the connected printer as "CLIENT DEFINED" (using Label Design Mode) on the Server as described in Chapter 2 of the LLM User's Guide and Chapter 2 of this guide.
 - Design a label format in Label Design Mode. Make sure the label that is designed prints successfully from Label Design Mode.

- Once all the above steps have been followed, in the Sample Programs subdirectory of Loftware, locate the Internet Active X Control Sample 1 program.
 - Either double click on **InternetActiveControlSample1.exe** or if you have Visual Basic, open the program in Visual Basic by double clicking on the .vbp file. Follow the steps that are displayed on the form to successfully print a label. The items in Blue are the associated properties, methods or events associated with the object it is next to. The items in red are mandatory fields that must be filled with the correct data.
2. Create a project in a programming tool that can utilize ActiveX Controls.
 3. Add the **Loftware Internet ActiveX Control** to the new project
 4. Connect to the Web Server by invoking the Connect Method.
 5. Select a printer by invoking the SetPrinter Method.
 6. Choose a label by invoking the SetLabelName Method.
 7. Set field data by using the SetData Method.
 8. Print the label format by using the PrintJob Method.

Helpful Hints and Common Mistakes

Note: 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. (See the Clustering chapter for more information.) Many of the iX methods throw critical errors during the failover transition. See the Error Event section at the end of this chapter for more information.

- Make sure that any label files created with versions of Loftware less than 3.5.2.58 are re-saved. The control uses a list inside the label format that is generated when the label is saved in versions 3.5.2.58 and newer. 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.

Design Scenario and Distribution

Overview

The iX control can be integrated into the user or integrator's applications running on remote systems. The Loftware Print Server (LPS) is installed at a central location driving the connected clients through the Loftware 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 free of charge. 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 free of charge. 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 is a list of the files that are needed to use the control. These files are found in the `redist` directory under the Internet ActiveX 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.

atl.dll (for NT) v.3.00.8141 or greater ->winsys folder. This file needs to be registered.

atl.dll (for 95/98) v.2.00.7274 or greater->winsys folder. This file needs to be registered.

Internet ActiveX Properties

Loftware's ActiveX Properties are as follows:

Duplicates Property	FieldCount Property	FieldLength Property
FieldName Property	isConnected Property	JobName Property
Pages Property	PrinterAlias Property	PrinterCount Property
PrinterName Property	PrinterNumber Property	PrinterPort Property
PrinterTimeout Property	Quantity Property	TrimLeadingSpaces Property
WebAddress Property	WebPort Property	WebUserName Property

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
```

See also

Quantity Property	Pages Property
-----------------------------------	--------------------------------

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
```

See also

FieldName	FieldLength
-----------	-------------

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
```

See also

FieldName	FieldCount
-----------	------------

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
```

See also

FieldCount

FieldLength

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
```

See also

Quantity	Duplicates
----------	------------

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
```

See also

PrinterName	PrinterCount	PrinterPort
-------------	--------------	-------------

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 Loftware 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
```

See also

PrinterName	PrinterAlias	PrinterPort
-----------------------------	------------------------------	-----------------------------

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 Loftware 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
```

See also

PrinterAlias

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
```

See also

SetLabelName	SetData	PrintJob
------------------------------	-------------------------	--------------------------

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
```

See also

[PrinterName](#)

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
```

See also

Duplicates	Pages
------------	-------

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.lwl"  
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	ClearAllData Method	Connect Method
Disconnect Method	PrintJob Method	ResetJob Method
SetData Method	SetLabelName Method	SetPrinter Method
AppendJob Method	ClearAllData Method	

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
```

See also

PrintJob	SetData	SetLabelName
----------	---------	--------------

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
```

See also

ResetJob

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 PAS 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

```
CWebX1.Quantity = txtQuantity.Text  
CWebX1.Duplicates = txtDuplicates.Text  
CWebX1.PrintJob
```

See also

SetLabelName	SetData	AppendJob
------------------------------	-------------------------	---------------------------

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  
CWebX1.ClearAllData  
'Removes all queued jobs  
CWebX1.ResetJob  
End Sub
```

See also

ClearAllData

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
```

See also

SetLabelName	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
```

See also

PrintJob

SetData

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
```

See also

PrinterTimeout	PrinterPort
----------------	-------------

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	BeforePrint Event	ErrorEvent Event
InfoEvent Event	OtherEvent Event	PrintTimeout Event
WarningEvent Event		

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 with version 3.5.2.58 of Ilmwdn32.exe or greater.
26504	Label not set	An attempt to access a field or print a job with having called 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.

ID	Message	Explanation
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.

See also

WarningEvent	OtherEvent	InfoEvent
--------------	------------	-----------

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
```

See also

WarningEvent	ErrorEvent	InfoEvent
--------------	------------	-----------

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.

See also

ErrorEvent	OtherEvent	InfoEvent
----------------------------	----------------------------	---------------------------

Internet ActiveX Reference Table

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

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

Chapter 7 Loftware .NET Control

Overview of Loftware .NET Control

If you are already using .NET, you will find that with Loftware's .NET Control, you can easily integrate Loftware's bar code printing modules directly into your own applications, just as you have done with Loftware's previous Controls - the ActiveX System Control, the ActiveX Client Control, and the Internet ActiveX Control.

These controls diminish the level of knowledge and expertise required to connect and print to stand-alone and networked bar code label printers. 32-bit development languages such as C#, J#, managed C++, and VB.NET can utilize this innovative technology as long as they are a CLR compliant language.

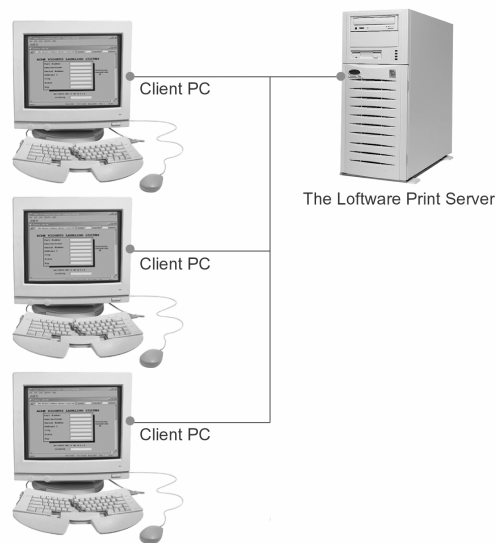


Figure 7-A: PCs using one of the Client Controls to offload print jobs to the LPS

This chapter documents the Loftware .NET Control. Like the other Loftware Client Controls, this control 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 a "Client Control" because it acts as a client to the LPS (Loftware Print Server). You can use this control when your application is running in several places, needs to access many printers, and requires a small footprint.

The Loftware .NET Control utilizes the Microsoft .NET© framework. This control is designed for use in .NET applications where your current ActiveX Client Controls may not be suitable.

Factors with the .NET Control

The Loftware .NET Control acts similar to the other Loftware Client Controls. However, for this new control, the file drop functionality is not implemented. A client connection from the control is cleaner, easier to control, and more scaleable. A list of factors in the use of the .NET Control follows:

- The LPS is logged in to as a client application.
- The .NET Control uses the *.XML file type for print requests.
- Print requests are created programmatically by providing data for each field on the selected label.
- Quantity, duplicates, pages, and field data are set for each label.
- The Print job or Loaded label only exists in memory while the client is logged in. All of the print jobs are deleted from memory and the label is unloaded if logged out.
- One label at a time may be loaded.

Installation and Use of the Loftware .NET Control

Pre-Install Information

Requirements

- Microsoft Visual Studio .NET 2003 is required to open and build any of the Loftware .NET Control sample solutions listed below.
- Microsoft .NET Framework 1.1 is required for development using the Loftware .NET Control.

Note: If you do not already have the Microsoft .NET Framework 1.1 installed, it is installed during the Loftware .NET Control install.

- Microsoft Internet Explorer 5.01 or later is required to install the Microsoft .NET Framework 1.1.
- One of the following Operating Systems is required:
 - Microsoft Windows 2000 Professional
 - Microsoft Windows 2000 Server
 - Microsoft Windows 2000 Advanced Server
 - Microsoft Windows XP Home Edition
 - Microsoft Windows XP Professional
 - Microsoft Windows 2003

NOTE: .NET installation on Windows 2000 requires Service Pack 3.
- Pentium 3 600 MHz or faster is required.
- 64 MB required but 128 MB or higher recommended.

Sample Programs and Read Me Files

Before attempting to use the Loftware .NET Control, please review the sample programs and their associated Read Me files. The sample programs can be found at:

C:\Program Files\Loftware Labeling\Loftware .NET Control\Samples... assuming you installed Loftware to your C drive.

There are 3 sample programs in this folder:

C# Samples

1. TestClientSimple - This project is an implementation of Loftware's .NET Control written in C#. Its purpose is to show you how to use the majority of the methods that Loftware's .NET Control offers.
2. TestClientAdvanced - The TestClientCSharp project is an implementation of the .NET Control written in C#. It is based on Loftware's On Demand Print Client and the Web Client.

VB .NET Samples

3. Loftware .NET Control VB Test Client - The VBTestClient project is an implementation of Loftware's .NET Control written in VB.NET. Its purpose is to show you how to use the majority of the methods that Loftware's .NET Control offers.

The sample programs' Read Me files contain build steps and special notes to help you understand the Loftware .NET Control. Please take a few moments to review them first, as this may save you a lot of time in the long run.

Pre-Installation

- Install the Loftware Print Server (LPS). Refer to Chapter 1 for installation instructions. Please note that the LPS can be installed on any NT/2000/XP computer anywhere on your LAN or WAN.
- Install the Loftware .NET Control. (See the following.)

Installing the Loftware .Net Control

Installing from the CD

If you have installed Loftware from the CD, a folder called **Loftware .NET Installs** has already been created under Loftware Labeling.

1. Open the folder below called **Loftware .NET Control Install**.
2. Double-click on the **setup.exe** application.

The first part of the install checks the system for Microsoft Windows® Installer version 2.0 or higher. If this version or higher is not found on the machine, the Loftware .NET Control install updates the Windows installer version. A reboot may be required depending on the operating system.

The second part of the install checks the system for the existence of the Microsoft® .NET framework version v1.1. If this version does not exist, the .Net framework is installed on the machine. A series of screen prompts are displayed. The install of the .NET framework may take up to 10 minutes depending on the machine/operation system and it may also require a reboot.

The third part of the install installs the Loftware .NET Control application and samples to:
Program Files > Loftware Labeling > Loftware .NET Control.

- a. Press **Next** at the Welcome screen.
- b. Choose "Complete" or "Custom" Install, and press **Next** at the Setup Type screen.

- c. Press **Install** at the Ready to Install screen.
 - d. Press **Finish** at the Install Shield Wizard Completed screen.
3. Configure all connected printers (using Label Design Mode) on the Server as described in Chapter 2 of the LLM User's Guide.
 4. Design the necessary label formats (using Label Design Mode) on the server.
 5. You may also design them on any client PC and either share the files on the server or copy the files to the server when done. The **Options | File Locations** menu in label design mode allows you to set network locations for your design files. This allows you to design on a client PC while your labels are saved to the PC specified...usually the LPS PC
 6. Verify that the LPS is working by building a simple label and manually creating a .pas file for it. Copy your pas file to the directory that LPS is scanning and your label should print. If it does not, refer to Chapter 1 for ideas on troubleshooting.

Using the Loftware .NET Control

1. Install the Loftware .NET Control on a client PC.
2. Load one of the sample programs as mentioned above; learn and study it. (This saves you time in the long run.)

The sample programs for the Loftware .NET Control are found under:

C:\Program Files\Loftware Labeling\Loftware .NET Control\Samples, assuming that C:\ is the install directory for Loftware.

3. Create a new project using Microsoft Visual Studio 2003.
4. Add the Loftware .NET Control to the new project. For example, follow these steps if you are using Microsoft Visual Studio .NET 2003:
5. Add a reference to the Loftware.LLMControl.dll.
 - a. Right click on **References** in Solution Explorer and select Add Reference.
 - b. Select the .NET tab (selected by default) in the Add Reference dialog, click **Browse** and browse to “..\Loftware Labeling\Loftware .NET Control”.
 - c. Select the Loftware.LLMControl.dll, click **Open**, and then click **OK** in the Add Reference dialog. A reference to Loftware.LLMControl is now shown under the References in your project.

If your project is a **Windows Application**, follow these steps:

- a. Open the main form of your project in design view.
- b. Open the Toolbox and select the **My User Controls** tab. Right click and choose Add/Remove Items.
- c. Select the .NET Framework Components tab (selected by default) in the Customize Toolbox dialog, click **Browse** and browse to “..\Loftware Labeling\Loftware .NET Control”.
- d. Select the Loftware.LLMControl.dll and click **Open**. LoftClient is then added to the list of .NET Framework Components.
- e. Verify that the check box next to LoftClient is checked (checked by default) and click **OK**.

- f. Select the LoftClient component under **My User Controls** and place it on the main form of your project. An item named 'LoftClient1' is displayed in a window below your form.
- g. Right click on it and choose properties if you want to change its default name.

If your project is a **Console Application**, follow this step:

During the initialization of your application, create an instance of the Software .NET Control. For example:

```
Software.LLMControl.LoftClient m_LoftClient = new  
Software.LLMControl.LoftClient();
```

IMPORTANT NOTE: The Software.LLMControl.DLL must be in the same directory as the executable for your project. Copy and paste the DLL to your debug, release, or bin folders as necessary.

Distributing the Software .NET Control

The Software .NET Control can be distributed with your application(s) free of charge. When building an install program for your application, include the required Software.LLMControl.dll with it. Your setup program should take care of installing the Software.LLMControl.dll, thereby avoiding having to run two setup programs.

Note: With Version 8.4 and above, all the required Software DLLs have been consolidated into one – the Software.LLMControl.dll.

Using the .NET Control in a Web Application

It is possible to use the Software .NET Control in a web application such as ASP.NET. To do so the project must contain a Web.config file. To add a Web.config file to a Microsoft Visual Studio .NET 2003© project follow these steps:

1. Click on the "Project" menu.
2. Select "Add New Item..."
3. In the Add New Item dialog, select "Web Configuration File."
4. Click Open.

The Web.config file is an XML file. To use the .NET Control in a web application it is necessary to edit this file. Open it and add the following to the <configuration> section:

```
<appSettings>  
  <add key="assemblyPath" value="C:\TEMP\dotNetControl"/>  
</appSettings>
```

The assemblyPath value can be set to any directory but the directory must contain the Software.LLMControl.dll.

The DLL can reside anywhere* as long as the assemblyPath value points to its location. The Software.LLMControl.dll must reside in the same directory as the web application.

Note: Even if the `Loftware.LLMControl.dll` is in the same directory as the web application, the `Web.Config` file still needs to exist with the `assemblyPath` value pointing to the directory of the web application.

Loftware .NET Control Methods

Methods are used when the integrator wants to get information, format information, or send information. If you see a method definition with two parameter sets, that method is overloaded. The methods described in this section are used in the Loftware .NET Control.

AppendJob	Clear Data	GetFieldIndex	GetFieldMaxLength	GetFieldName
GetLabel	GetLabelList	GetLabelListExt	GetPrinterByNumber	GetPrinters
KillJobs	Login	Logout	PrintJob	SetData

AppendJob

Declaration

```
Public Sub AppendJob()
```

Syntax

```
control.AppendJob()
```

Description

AppendJob takes the current job in memory and saves it on a list, allowing you to work on a new job and still be able to “batch print” the other jobs.

Example

```
Private Sub Btnappend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnappend.Click
    LoftClient.PrinterName = "dotnetprinter"
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of
the field
    LoftClient.SetData(1, TextBox2.Text) '1 is the index number of the
field
    LoftClient.AppendJob()'Appends this job to the queue
End Sub
```

ClearData

Declaration

```
Public Sub ClearData()
```

Syntax

```
control.ClearData()
```

Description

The ClearData method clears all of the user-entered data from the current job.

Example

```
Private Sub Btnappend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnappend.Click
    LoftClient.PrinterName = "dotnetprinter"
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 = name of the field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.ClearData() 'Clears the data that was entered for current job
End Sub
```

GetFieldIndex

Declaration

```
Public Function GetFieldIndex(ByVal strFieldName As String) As Integer
```

Syntax

```
nFieldIndex = control.GetFieldIndex(strFieldName)
```

Description

The GetFieldIndex method allows you to obtain the index of the field based on the field name.

Example

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim iCount As Integer
    Dim lblFieldTest As String
    LoftClient.Login("172.16.0.93", 2723)
    ListLabels("")
    printgrid()
    LoftClient.GetLabel("c:\Program files\Software Labeling\Labels\dotnet.lwl")
    LoftClient.LabelDtd = ("c:\Program files\Software
        Labeling\batch\label.dtd")
    lblgrid()
    lblindex0.Text = LoftClient.GetFieldIndex("Text0000").ToString()
    lblindex1.Text = LoftClient.GetFieldIndex("Text0001").ToString()
    lblserver.Text = LoftClient.CurrentServer
End Sub
```

GetFieldMaxLength

Declarations

```
Public Function GetFieldMaxLength(ByVal iFieldNum As Integer) As Integer  
Public Function GetFieldMaxLength(ByVal strFieldName As String) As Integer
```

Syntax

```
nMaxFieldLen = control.GetMaxFieldLength(strFieldName)
```

or

```
nMaxFieldLen = control.GetMaxFieldLength(nFieldIndex)
```

Description

The GetFieldMaxLength method takes the Index or Name of the fields and returns the maximum number of allowed characters for that field.

Example

```
' gets user entered data from the list view and puts it into an edit box  
' restricts the amount of data a user can type into a edit box  
Private Sub lstLabelData_SelectedIndexChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
lstLabelData.SelectedIndexChanged  
    Dim iMaxFieldLen As Integer  
    'get the selected item's text, put it into the data entry edit box  
    If lstLabelData.SelectedIndices.Count Then  
        ' get the field max length so we can constrain input  
        iMaxFieldLen=mLlmClient1.GetFieldMaxLength(lstLabelData.SelectedIndices.  
            Item(0))  
        DataEntry.MaxLength = iMaxFieldLen  
        If lstLabelData.SelectedItems(0).Text.Length() Then  
            DataEntry.Text = lstLabelData.SelectedItems(0).Text  
        End If  
    End If  
End Sub
```

GetFieldName

Declaration

```
Public Function GetFieldName(ByVal iFieldNum As Integer) As String
```

Syntax

```
strFieldName = control.GetFieldName(nFieldIndex)
```

Description

The GetFieldName method takes the Index of the field and returns that Name. This function allows you to quickly iterate through all the fields after loading a label using the FieldCount property (next section).

Example

```
' get the label information and add it to the list view
Private Sub GetFields()
    Dim i As Integer
    Dim Item As ListViewItem
    Dim sFieldName As String
    Dim iMaxFieldLen As Integer
    ' get the label whose name is stored in mLabelName
    mLlmClient1.GetLabel(mLabelName)
    For i = 0 To mLlmClient1.FieldCount - 1
        Item = New ListViewItem("")
        ' get the field name from the control
        sFieldName = mLlmClient1.GetFieldName(i)
        ' get the max field length from the control
        iMaxFieldLen = mLlmClient1.GetFieldMaxLength(i)
        ' add the field name and max field length to the list view
        Item.SubItems.Add(sFieldName)
        Item.SubItems.Add(System.Convert.ToString(iMaxFieldLen))
        lstLabelData.Items.Add(Item)
    Next
    ' resize the list view columns to fit
    lstLabelData.Columns(0).Width = lstLabelData.Width / 3
    lstLabelData.Columns(1).Width = lstLabelData.Width / 3
    lstLabelData.Columns(2).Width = lstLabelData.Width / 3
End Sub
```

GetLabel

Declaration

```
Public Function GetLabel(ByVal strLabelName As String) As Boolean
```

Syntax

```
bResult = control.GetLabel(strLabelName)
```

Description

The GetLabel method downloads the label information from the LPS and sets it as the currently loaded label in memory. This method must be called before you begin work on a job.

Example

```
' get the label information and add it to the list view
Private Sub GetFields()
    Dim i As Integer
    Dim Item As ListViewItem
    Dim sFieldName As String
    Dim iMaxFieldLen As Integer
    ' get the label whose name is stored in mLabelName
    mLlmClient1.GetLabel(mLabelName)
    For i = 0 To mLlmClient1.FieldCount - 1
        Item = New ListViewItem("")
        ' get the field name from the control
        sFieldName = mLlmClient1.GetFieldName(i)
        ' get the max field length from the control
        iMaxFieldLen = mLlmClient1.GetFieldMaxLength(i)
        ' add the field name and max field length to the list view
        Item.SubItems.Add(sFieldName)
        Item.SubItems.Add(System.Convert.ToString(iMaxFieldLen))
        lstLabelData.Items.Add(Item)
    Next
    ' resize the list view columns to fit
    lstLabelData.Columns(0).Width = lstLabelData.Width / 3
    lstLabelData.Columns(1).Width = lstLabelData.Width / 3
    lstLabelData.Columns(2).Width = lstLabelData.Width / 3
End Sub
```

GetLabelList

Declaration

```
Public Function GetLabelList(ByVal strSubDir As String, ByRef strLabelList()
As String) As Boolean
```

Syntax

```
bResult = control.GetLabelList(strFolder, arrLabelNames)
```

Description

The GetLabelList method returns a list of all available labels on the LPS as a string array. You may pass in a path if you wish to specify another path on the LPS for the labels directory. No path defaults to the current label directory.

Example

```
' gets a list of available labels and displays them in a listbox
Private Sub GetLabels()
    Dim LabelList() As String
    Dim Folder As String
    Dim ErrorBox As MessageBox
    '
    lstLabels.DataSource = Nothing
    ' get a list of available labels
    If LlmClient1.GetLabelList(Folder, LabelList) <> True Then
        ErrorBox.Show("Failed to get list of labels from server")
        Return
    End If
    lstLabels.DataSource = LabelList
End Sub
```

GetLabelListExt

Declaration

```
Public Function GetLabelListExt(ByVal strSubDir As String, ByRef
strLabelList() As String) As Boolean
```

Syntax

```
bResult = control.GetLabelListRxt(strFolder, arrLabelNames)
```

Description

The GetLabelListExt method is similar to the GetLabelList method, except that GetLabelListExt returns a list of with file extensions on the LPS as a string array. You may pass in a path if you wish to specify another path on the LPS for the labels directory. No path defaults to the current label directory.

Example

```
' gets a list of available labels and displays them in a listbox
Private Sub GetLabelsAndTags()
    Dim LabelList() As String
    Dim Folder As String
    Dim ErrorBox As MessageBox
    lstLabels.DataSource = Nothing
    ' get a list of available labels with file extensions
    If LlmClient1.GetLabelListExt(Folder, LabelList) <> True Then
        ErrorBox.Show("Failed to get list of labels from server")
        Return
    End If
    lstLabels.DataSource = LabelList
End Sub
```

GetPrinterByNumber

Declaration

```
Public Function GetPrinterByNumber(ByVal iPrinterNumber As Integer, ByRef  
strPrintName As String, ByRef strPrintAlias As String, ByRef strPrintPort As  
String) As Boolean
```

Syntax

```
bResult = control.GetPrinterByNumber(nPrinterNumber, strPrinterName,  
strPrinterAlias, strPrinterPort)
```

Description

The `GetPrinterByNumber` method takes in the LLM Printer number and gives back the name, alias (if any) and port of the specified printer as strings.

Example

```
' gets a list of available printers and displays them in a list view  
Private Sub GetPrinters()  
    Dim sPrinter As String  
    Dim sAlias As String  
    Dim sPort As String  
    Dim i As Integer  
    Dim ErrorBox As MessageBox  
    Dim Item As ListViewItem  
    lstPrinters.Items.Clear()  
    ' get a list of available printers  
    If LlmClient1.GetPrinters(mPrinterNumbers) <> True Then  
        ErrorBox.Show("Failed to get printer list")  
        Return  
    End If  
    For i = 0 To mPrinterNumbers.Length - 1  
        ' get the printer name, alias and port  
        If LlmClient1.GetPrinterByNumber(mPrinterNumbers(i), sPrinter, sAlias,  
            sPort) <> True Then  
            ErrorBox.Show("Failed to get printer name")  
            Return  
        Else  
            Item = New ListViewItem(" " + System.Convert.ToString(mPrinterNumbers(i)) +  
                " ")  
            Item.SubItems.Add(sPrinter)  
            Item.SubItems.Add(sAlias)  
            Item.SubItems.Add(sPort)  
            lstPrinters.Items.Add(Item)  
        End If  
    Next  
    ' set the width of the listview columns  
    lstPrinters.Columns(0).Width = lstPrinters.Width / 4  
    lstPrinters.Columns(1).Width = lstPrinters.Width / 4  
    lstPrinters.Columns(2).Width = lstPrinters.Width / 4  
    lstPrinters.Columns(3).Width = lstPrinters.Width / 4  
End Sub
```

GetPrinters

Declaration

```
Public Function GetPrinters(ByRef iPrinterList() As Integer) As Boolean
```

Syntax

```
bResult = control.GetPrinters(arrPrinterNumbers)
```

Description

The `GetPrinters` method obtains a list of all configured printers from the LPS and returns a list of the configured printer numbers as an `int []`. This `int []` should be used to obtain the printer information using the `GetPrinterByNumber` function call.

Example

```
' gets a list of available printers and displays them in a listview
Private Sub GetPrinters()
    Dim sPrinter As String
    Dim sAlias As String
    Dim sPort As String
    Dim i As Integer
    Dim ErrorMessage As MessageBox
    Dim Item As ListViewItem
    lstPrinters.Items.Clear()
    ' get a list of available printers
    If LlmClient1.GetPrinters(mPrinterNumbers) <> True Then
        ErrorMessage.Show("Failed to get printer list")
        Return
    End If
    For i = 0 To mPrinterNumbers.Length - 1
        ' get the printer name, alias and port
        If LlmClient1.GetPrinterByNumber(mPrinterNumbers(i), sPrinter, sAlias,
            sPort) <> True Then
            ErrorMessage.Show("Failed to get printer name")
            Return
        Else
            Item = New ListViewItem(" " +
                System.Convert.ToString(mPrinterNumbers(i)) + " ")
            Item.SubItems.Add(sPrinter)
            Item.SubItems.Add(sAlias)
            Item.SubItems.Add(sPort)
            lstPrinters.Items.Add(Item)
        End If
    Next
    ' set the width of the listview columns
    lstPrinters.Columns(0).Width = lstPrinters.Width / 4
    lstPrinters.Columns(1).Width = lstPrinters.Width / 4
    lstPrinters.Columns(2).Width = lstPrinters.Width / 4
    lstPrinters.Columns(3).Width = lstPrinters.Width / 4
End Sub
```

KillJobs

Declaration

```
Public Sub KillJobs (ByVal bUnloadLabel As Boolean)
```

Syntax

```
control.KillJobs(True)
```

Description

The KillJobs method removes all the jobs from memory (saved via AppendJob) and clears the job you are currently working on. If you pass in true to the parameter, then KillJobs also “unloads” the currently loaded label from memory.

Example

```
Private Sub btnkillJobs_Click (ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnkillJobs.Click  
    LoftClient.KillJobs(False) 'clears jobs from the queue if set to true it  
        will also unload the label from memory  
End Sub
```

Login

Declaration

```
Public Function Login (ByVal strAddr As String, ByVal iPort As Integer) As  
Boolean
```

Syntax

```
bResult = control.Login(strIpAddress, nPort)
```

Description

The Login method takes the server name or IP address (string) and port number (int) and attempts to log in as a client to the LPS.

Example

```
' logs the user in or out in response to clicking the login button  
Private Sub btnLogin_Click (ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnLogin.Click  
    ' if we're logged in, log us out  
    If LlmClient1.LoggedIn() Then  
        btnLogin.Text = "Login"  
        LlmClient1.Logout()  
        lstPrinters.Items.Clear()  
        lstLabels.DataSource = Nothing  
    Else ' otherwise, log us in  
        btnLogin.Text = "Logout"  
        LlmClient1.Login(IPAddress.Text, 2723)  
        GetLabels()  
        GetPrinters()  
    End If  
End Sub
```

Logout

Declaration

```
Public Sub Logout()
```

Syntax

```
control.Logout()
```

Description

The Logout method sends a logout request to the LPS and then closes out the connected socket.

Example

```
'Sends a logout request to the LPS and then closes out the connected socket.
Private Sub btnlogout_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnlogout.Click
    LoftClient.Logout()
End Sub
```

PrintJob

Declaration

```
Public Function PrintJob(ByRef prtJobRsp As
Loftware.LLMControl.PrintJobResponse) As Integer
```

Syntax

```
nJobNumber = LoftClient1.PrintJob(prtJobRsp)
```

or, if you don't want data back:

```
nJobNumber = LoftClient1.PrintJob()
```

Description

The PrintJob method prints all of the jobs in memory, including the one currently being worked on. When PrintJob returns, the optional prtJobRsp object contains data that can be used to determine the result of the submitted job. See *Class PrintJobResponse* for more information.

Example

```
Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnPrint.Click
    Dim prtJobRsp As Loftware.LLMControl.PrintJobResponse
    Dim nJobNum As Integer
    LoftClient1.GetLabel("TestLabel.lwl")
    LoftClient1.SetData("TestField", "TestData")
    LoftClient1.PrinterNumber = 1
    nJobNum = LoftClient1.PrintJob(prtJobRsp)
    'nJobNum will be 0 if it's a 'critical error'
    'therefore, if it is not 0, check the StatusCode before trying
```

```

'to grab data
  if ((nJobNum <> 0) And (prtJobRsp.StatusCode = 4))
    m_strXmlData = prtJobRsp.XmlData
'now parse whatever data you want out of the XML
  Else
    MessageBox("Errors Encountered", MsgBoxStyle.Critical)
  End If
End Sub

```

Class PrintJobResponse

Declaration

```
Public Class PrintJobResponse
```

Syntax

```
Dim prtJobRsp As Software.LLMControl.PrintJobResponse
```

Description

The PrintJobResponse class is passed into the PrintJob method to obtain valuable data from the LPS in response to a print job request. At the time of this writing, the only data that is passed back is the EPC data related to RFID printing, and pass/fail job status. Future versions will have the ability to pass back other kinds of data. This is a synchronous call and may be blocked if the LPS is under a very heavy load. In other words, your code will not continue until the LPS has printed the request. For this reason, we suggest placing your print job code in a separate thread so that your program can be doing other things in the event it has to wait.

The PrintJobResponse class contains the following properties:

public int StatusCode	Each print job request results in one of the following status codes: 2 = Critical Failure 4 = Printed 6 = Printed with Errors 8 = Printer Error
public string StatusMessage	Each print job request results in one of the following status messages: "Critical Failure" "Printed" "Printed with Errors" "Printer Error"
public string ErrorMessage	Contains the error message if the print job request encountered any errors; otherwise, it is empty.
public int PrinterNumber	Printer number of the print job request
public int JobNumber	Job number of the print job request assigned by the LPS
public object Reserved	Empty, reserved for future use
public string XmlData	Response to the print job request from the LPS in XML form. May contain EPC data, see the following examples of possible XmlData values.

Note: To properly parse the XML response data, the LoftDataResp.xsd file must be copied to the path containing the executable file of the application that uses the .NET Control. If the XML responses are saved to disk, the LoftDataResp.xsd file must be copied to the same directory the XML responses are saved to. The LoftDataResp.xsd file is installed to the "Loftware Labeling\Batch" directory.

Example 1 – Single request containing EPC data

```
<?xml version="1.0" encoding="utf-8" ?>
<job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LoftDataResp.xsd"
name="20041119153416000011709_RKNOWLES-XP2" jobnumber="11709" device="6">
  <status code="4" text="" />
  <request number="1" LabelNumber="1">
    <field name="RFID" value="B80F600390000064">
      <property name="EncodingType" value="SGTIN-64"/>
      <property name="FilterValue" value="7"/>
      <property name="CompanyPrefixIndex" value="123"/>
      <property name="ItemReference" value="456"/>
      <property name="SerialNumber" value="100"/>
    </field>
  </request>
</job>
```

Example 2 – Stacked request containing EPC data

```
<?xml version="1.0" encoding="utf-8" ?>
<job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LoftDataResp.xsd"
name="20041119153416000011709_RKNOWLES-XP2" jobnumber="11709" device="6">
  <status code="4" text="" />
  <request number="1" LabelNumber="1">
    <field name="RFID" value="B80F600390000064">
      <property name="EncodingType" value="SGTIN-64"/>
      <property name="FilterValue" value="7"/>
      <property name="CompanyPrefixIndex" value="123"/>
      <property name="ItemReference" value="456"/>
      <property name="SerialNumber" value="100"/>
    </field>
  </request>
  <request number="1" LabelNumber="2">
    <field name="RFID" value="B80F600390000065">
      <property name="EncodingType" value="SGTIN-64"/>
      <property name="FilterValue" value="7"/>
      <property name="CompanyPrefixIndex" value="123"/>
      <property name="ItemReference" value="456"/>
      <property name="SerialNumber" value="101"/>
    </field>
  </request>
```

```

<request number="1" LabelNumber="3">
  <field name="RFID" value="B80F600390000066">
    <property name="EncodingType" value="SGTIN-64"/>
    <property name="FilterValue" value="7"/>
    <property name="CompanyPrefixIndex" value="123"/>
    <property name="ItemReference" value="456"/>
    <property name="SerialNumber" value="102"/>
  </field>
</request>
</job>

```

Example 3 – Single request without EPC data

```

<?xml version="1.0" encoding="utf-8" ?>
<job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LoftDataResp.xsd"
name="20041201100415000011798_RKNOWLES-XP2" jobnumber="11798" device="1">
  <status code="2" text="Pass File Error" />
</job>

```

SetData

Declaration

```
Public Sub SetData(ByVal iFieldToSet As Integer, ByVal strData As String)
```

Syntax

```
control.SetData(nFieldIndex, strDataValue)
```

or

```
control.SetData(strFieldName, strDataValue)
```

Description

The SetData method takes the name or index of the field the user wishes to set the data for as well as the data as a string.

Example

```

Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
  LoftClient.GetLabel("c:\Program files\Loftware Labeling\Labels\dotnet.lwl")
  LoftClient.PrinterNumber = 1
  LoftClient.JobName = "job2723"
  LoftClient.Quantity = 1
  LoftClient.Duplicates = 1
  LoftClient.Pages = 1
  LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
  field
  LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
  LoftClient.PrintJob()
End Sub

```

Loftware .NET Control Properties

This section describes Loftware .NET Control properties and their access abilities (get, set, or both). The Loftware .NET Control properties refer to the currently loaded label.

CurrentLabel	CurrentServer	Duplicates	FieldCount	JobName
LabelDtd	LastErrorMessage	LoggedIn	Pages	PrinterName
PrinterNumber	Quantity	Tray		

CurrentLabel

Declaration

```
Public ReadOnly Property CurrentLabel() As String
```

Syntax

```
strCurLabel = control.CurrentLabel
```

Description

The CurrentLabel property maps directly to the currently loaded label. This is only valid after calling GetLabel().

Example

```
Private Sub lblgrid() '(ByVal sender As System.Object, ByVal e As System)
    lbllabel.Text = LoftClient.CurrentLabel
End Sub
```

CurrentServer

Declaration

```
Public ReadOnly Property CurrentServer() As String
```

Syntax

```
strServer = control.CurrentServer
```

Description

The CurrentServer property is the name or IP address of the LPS to which you are currently connected.

Example

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
LoftClient.Login("172.16.0.93", 2723)
    ListLabels("")
    printgrid()
```

```
        LoftClient.GetLabel("c:\Program files\Software  
        Labeling\Labels\dotnet.lwl")  
    LoftClient.LabelDtd = ("c:\Program files\Software Labeling\batch\label.dtd")  
    lblgrid()  
    lblserver.Text = LoftClient.CurrentServer  
End Sub
```

Duplicates

Declaration

```
Public Property Duplicates() As Integer
```

Syntax

```
control.Duplicates=1
```

Description

The Duplicates property allows you to both get and set the number of duplicates you want for the current job. This is only valid after calling GetLabel().

Example

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
    LoftClient.Duplicates=1  
End Sub
```

FieldCount

Declaration

```
Public ReadOnly Property FieldCount() As Integer
```

Syntax

```
nFieldCount = control.FieldCount
```

Description

The FieldCount property allows you to get the total number of fields in the currently loaded label. This is only valid after calling GetLabel().

Example

```
' get the label information and add it to the list view
Private Sub GetFields()
    Dim i As Integer
    Dim Item As ListViewItem
    Dim sFieldName As String
    Dim iMaxFieldLen As Integer
    ' get the label whose name is stored in mLabelName
    mLlmClient1.GetLabel(mLabelName)
    For i = 0 To mLlmClient1.FieldCount - 1
        Item = New ListViewItem("")
        ' get the field name from the control
        sFieldName = mLlmClient1.GetFieldName(i)
        ' get the max field length from the control
        iMaxFieldLen = mLlmClient1.GetFieldMaxLength(i)
        ' add the field name and max field length to the list view
        Item.SubItems.Add(sFieldName)
        Item.SubItems.Add(System.Convert.ToString(iMaxFieldLen))
        lstLabelData.Items.Add(Item)
    Next
    ' resize the list view columns to fit
    lstLabelData.Columns(0).Width = lstLabelData.Width / 3
    lstLabelData.Columns(1).Width = lstLabelData.Width / 3
    lstLabelData.Columns(2).Width = lstLabelData.Width / 3
End Sub
```

JobName

Declaration

```
Public Property JobName() As String
```

Syntax

```
control.JobName = strJobName
```

Description

The JobName property allows you to get or set the Job Name for the current job you are working on. This is only valid after calling GetLabel().

Example

```
Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
    LoftClient.GetLabel("c:\Program files\Loftware Labeling\Labels\dotnet.lwl")
    LoftClient.PrinterNumber = 1
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
```

```

        LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
            field
        LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
        LoftClient.PrintJob()
    End Sub

```

LabelDtd

Declaration

```
Public Property LabelDtd() As String
```

Syntax

```
control.LabelDtd = strDtdPath
```

Description

The LabelDTD property allows you to get or set the current label.dtd path for the LPS. The label.dtd must exist in the same place on both the LPS and the Client in order for the xml creation to work properly.

Example

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    LoftClient.Login("172.16.0.93", 2723)
    LoftClient.LabelDtd = ("c:\Program files\Loftware Labeling\batch\label.dtd")
End Sub

```

LastErrorMessage

Declaration

```
Public ReadOnly Property LastErrorMessage() As String
```

Syntax

```
strLastError = control.LastErrorMessage
```

Description

The LastErrorMessage property allows you to obtain the last error message for the control. These error messages are the same as the ones that are captured by the ErrorMessage Event.

Example

```

Private Sub btnlasterrmsg_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnlasterrmsg.Click
    TextBox3.Text = LoftClient.LastErrorMessage
    If (TextBox3.Text = "") Then
        TextBox3.Text = "There has been no error messages"
    End If
End Sub

```

LoggedIn

Declaration

```
Public ReadOnly Property LoggedIn() As Boolean
```

Syntax

```
bLoggedIn = control.LoggedIn
```

Description

The LoggedIn property allows you to check if you are logged into the LPS or not. Also takes socket connection into consideration.

Example

```
' logs the user in or out in response to clicking the login button
Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
    ' if we're logged in, log us out
    If LlmClient1.LoggedIn() Then
        btnLogin.Text = "Login"
        LlmClient1.Logout()
        lstPrinters.Items.Clear()
        lstLabels.DataSource = Nothing
    Else ' otherwise, log us in
        btnLogin.Text = "Logout"
        LlmClient1.Login(IPAddress.Text, 2723)
        GetLabels()
        GetPrinters()
    End If
End Sub
```

Pages

Declaration

```
Public Property Pages() As Integer
```

Syntax

```
control.Pages=1
```

Description

The Pages property allows you to get or set the number of pages of labels printed. This refers to an entire copy of a page of labels when printing with layouts only.

Example

```
Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
    LoftClient.GetLabel("c:\Program files\Loftware Labeling\Labels\dotnet.lwl")
```

```

    LoftClient.PrinterNumber = 1
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
        field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.PrintJob()
End Sub

```

PrinterName

Declaration

```
Public Property PrinterName() As String
```

Syntax

```
control.PrinterName = strPrinterName
```

Description

Allows you to get or set the printer name (alias) of the printer to which you wish to print.

Example

```

Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
    LoftClient.GetLabel("c:\Program files\Software Labeling\Labels\dotnet.lwl")
    'LoftClient.PrinterNumber = 1
    LoftClient.PrinterName = "dotnetprinter"
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
        field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.PrintJob()
End Sub

```

PrinterNumber

Declaration

```
Public Property PrinterNumber() As Integer
```

Syntax

```
control.PrinterNumber=3
```


Description

The PrinterNumber property allows you to get or set the LLM printer number of the printer to which you wish to print.

Example

```
Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
    LoftClient.GetLabel("c:\Program files\Loftware Labeling\Labels\dotnet.lwl")
    LoftClient.PrinterNumber = 3
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
        field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.PrintJob()
End Sub
```

Quantity

Declaration

```
Public Property Quantity() As Integer
```

Syntax

```
control.Quantity=1
```

Description

The Quantity property allows you to get or set the total number of labels you want to print.

Example

```
Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnprint.Click
    LoftClient.GetLabel("c:\Program files\Loftware Labeling\Labels\dotnet.lwl")
    LoftClient.PrinterNumber = 1
    LoftClient.JobName = "job2723"
    LoftClient.Quantity = 1
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
        field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.PrintJob()
End Sub
```

Tray

Declaration

```
Public Property Tray() As String
```

Syntax

```
control.Tray = "Tray2"
```

Description

The Tray property allows you to set the name of the tray from which the stock is taken for the labels you want to print (Windows printers only).

Example

```
Private Sub Btnappend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnappend.Click
    LoftClient.PrinterName = "dotnetprinter"
    LoftClient.JobName = "job2723"
    LoftClient.Tray = "tray2"
    LoftClient.Duplicates = 1
    LoftClient.Pages = 1
    LoftClient.SetData("text0000", TextBox1.Text) 'text0000 is the name of the
        field
    LoftClient.SetData(1, TextBox2.Text) ' 1 is the index number of the field
    LoftClient.Tray() 'Selects the Tray Number for the Printer for current job.
End Sub
```

Loftware .NET Control Events

The Loftware .NET Control has three events within itself: ErrorMessage Event, WarningMessage Event, and InfoMessage Event. Each one signifies a different type of “error” message. Event handling is the primary and best way to handle errors when using this control for integration.

ErrorMessage Event

Declaration

```
Public Event ErrorMessage(ByVal strErrMsg As String, ByVal objSender As
Object)
```

Description

This event signifies some form of error has occurred. Both LPS errors (invalid logins, etc.) and printer error/critical failures may be returned through this event (which includes failure to generate proper XML for the LPS).

Example

```
Private Sub LoftClient_ErrorMessage(ByVal strErrMsg As String, ByVal objSender
As Object) Handles LoftClient.ErrorMessage
    MsgBox(strErrMsg)
End Sub
```

WarningMessage Event

Declaration

```
Public Event WarningMessage(ByVal strWarnMsg As String, ByVal objSender As Object)
```

Description

This event is thrown in the case of a warning condition. These conditions are things that are brought to your attention but are not necessarily critical. Printed with Errors falls into this category.

Example

```
Private Sub LoftClient_WarningMessage(ByVal strWarnMsg As String, ByVal objSender As Object) Handles LoftClient.WarningMessage  
    MsgBox(strWarnMsg)  
End Sub
```

InfoMessage Event

Description

This event is thrown for non-essential information. "Printed successfully" falls into this category.

Example

```
Private Sub LoftClient_InfoMessage(ByVal strInfoMsg As String, ByVal objSender As Object) Handles LoftClient.InfoMessage  
    MsgBox(strInfoMsg)  
End Sub
```

More Error Information and Return Codes

There are two other ways to handle errors in the Loftware .NET Control. The first, and most obvious, is return codes. Since most of our methods return some kind of return code, you should always check these codes first. The second way is via the `LastErrorMessage` property. This property allows you to obtain the last set error message from either the LPS or the Control.

Return Codes in .NET

Return codes from functions are almost always your first way to check error conditions while using any kind of API. In the Loftware .NET Control, it is important to keep a few basic ideas in place. The first is that the return codes are typically simple yes/no codes, which means that most of the return codes are of the Boolean type. In the case of the return code being Boolean, false always indicates failure, where as true always indicates success. The return codes for other functions are specified as needed.

LastErrorMessage

Whenever any type of critical error occurs within the Loftware .NET Control, the area that had the error calls `OnErrorMessage`, passing through an error string.

Reference Table for the Loftware .NET Control

Name	Get	Set	Type	Comment
<i>AppendJob</i>	N/A	N/A	Method	Batches label requests together.
<i>Clear Data</i>	N/A	N/A	Method	Clears all the user-entered data from the current job
<i>CurrentLabel</i>	X		Property	Maps directly to the currently loaded label.
<i>CurrentServer</i>	X		Property	Name or IP address of connected LPS.
<i>Duplicates</i>	X	X	Property	Gets and Sets the # of duplicates for the current job.
<i>ErrorMessage</i>	N/A	N/A	Event	Signifies some form of error has occurred
<i>FieldCount</i>	X		Property	Gets the total # of fields in the currently loaded label.
<i>GetFieldIndex</i>	X		Method	Obtain the index of the field based on the field name.
<i>GetFieldMax Length</i>	X		Method	Takes index or name of fields, returns the max # of allowed char's.
<i>GetFieldName</i>	X		Method	Takes index of field and returns the name
<i>GetLabel</i>	X		Method	Downloads label info from LPS sets it as current loaded label
<i>GetLabelList</i>	X		Method	Returns list of available labels on the LPS.
<i>GetLabelListExt</i>	X		Method	Returns list of available labels and RFID tags with file extensions
<i>GetPrinterByNumber</i>	X		Method	Takes in the LLM Printer #, gives back name, alias, port.
<i>GetPrinters</i>	X		Method	Obtains a list of all configured printers, returns as an integer.
<i>KillJobs</i>	N/A	N/A	Method	Removes all the jobs from memory, clears current job.
<i>InfoMessage</i>	N/A	N/A	Event	Thrown for non-essential information.
<i>JobName</i>	X	X	Property	Get or Set the current JobName for the current job.
<i>LabelDtd</i>	X	X	Property	Get or set the current label.dtd path for the LPS.
<i>LastErrorMessage</i>	X		Property	Obtain the last set error message from either the LPS or the Control.
<i>Login</i>	N/A	N/A	Method	Takes the Server Name or IP add. and port #, logs in as client to LPS.
<i>LoggedIn</i>	X		Property	Checks the user log in to the LPS
<i>Logout</i>	N/A	N/A	Method	Sends a logout request to the LPS and closes out connected socket.
<i>Pages</i>	X	X	Property	Get or set the number of pages of labels printed.
<i>PrintJob</i>	N/A	N/A	Method	Prints all of the current jobs
<i>PrinterName</i>	X	X	Property	Name of configured printer.
<i>PrinterNumber</i>	X	X	Property	Get or set LLM Printer Number to which you want to print.
<i>Quantity</i>	X	X	Property	Get or set the amount of labels to print.
<i>SetData</i>	X		Method	Takes name or index of field user wishes to use.
<i>Tray</i>	X	X	Property	Sets the tray number for the printer stock. (Windows Printers only)
<i>WarningMessage</i>	N/A	N/A	Event	Thrown for a warning condition.

Index

A

- Active/Active Cluster 4-2
- Active/Passive Cluster 4-2
- ActiveX Client
 - Design Scenario 5-3
 - Installing 5-2
 - Overview 5-1
 - Properties 5-5
- ActiveX Client Control
 - Reference Table 5-36
- ActiveX Client Control
 - Events
 - ErrorEvent 5-30
 - InfoEvent 5-33
 - OtherEvent 5-35
 - WarningEvent 5-34
- ActiveX Client Control
 - Methods
 - AppendJob 5-23
 - ClearData 5-24
 - Login 5-25
 - PrintJob 5-25
 - ResetJob 5-26
 - SetData 5-26
 - SetLabelName 5-28
 - SetServer 5-29
- ActiveX Client Control
 - Properties
 - ClientIniPath 5-5
 - DropDirectory 5-7
 - Duplicates 5-7
 - FieldCount 5-8
 - FieldLength 5-9
 - FieldName 5-10
 - isRunning 5-12
 - JobName 5-12
 - LabelsPath 5-13
 - LayoutPath 5-13
 - Pages 5-14
 - PasExt 5-19
 - PrinterAlias 5-14
 - PrinterCount 5-15

- PrinterName 5-16
- PrinterNumber 5-18
- PrinterPath 5-18
- PrinterPort 5-17
- Quantity 5-19
- ServerAlias 5-22
- ServerCount 5-20
- ServerName 5-21
- ServerNumber 5-21
- TimeLeadingSpaces 5-23

C

- Client Connections to LPS
 - Virtual Server 4-19
- Cluster Administrator 4-2
- Cluster Service 4-2
- Clustering 4-1

D

- Debugging 1-16

E

- Errors
 - Logging 1-16
- Event Log 1-46
- Event Logging 1-17

F

- Failover 4-3, 4-4
 - Conditions causing 4-4
 - Conditions that do not cause 4-4
 - Criteria 4-4
 - Manual Initiation 4-5
 - Retry before 4-4
- Field Lists 1-30
- File Drop 1-3
- File Share 4-9

I

- Installing
 - ActiveX Client 5-2
 - LPS to a Cluster 4-6
 - Modifying 4-18

- Interface 1-3
 - .NET 1-4
 - Connector 1-4
 - Direct Socket 1-4
 - File ActiveX 1-4
 - File Drop 1-3
 - File Web Client 1-4
 - On Demand Print Client 1-4
 - TCP/IP Socket 1-4
 - Internet ActiveX
 - Design Scenario 6-3
 - Overview 6-1
 - Reference Table 6-27
 - Internet ActiveX Events
 - AfterPrint 6-22
 - BeforePrint 6-22
 - ErrorEvent 6-22
 - InfoEvent 6-24
 - OtherEvent 6-25
 - WarningEvent 6-26
 - Internet ActiveX Methods
 - AppendJob 6-16
 - ClearAllData 6-17
 - Connect 6-17
 - Disconnect 6-18
 - PrintJob 6-18
 - ResetJob 6-18
 - SetData 6-19
 - SetLabelName 6-20
 - SetPrinter 6-20
 - TestConnection 6-21
 - TestPrinter 6-21
 - Internet ActiveX
 - Properties 6-5
 - Duplicates 6-5
 - FieldCount 6-5
 - FieldLength 6-6
 - FieldName 6-7
 - isConnected 6-8
 - JobName 6-8
 - Pages 6-9
 - PrinterAlias 6-10
 - PrinterCount 6-11
 - PrinterName 6-11
 - PrinterNumber 6-12
 - PrinterPort 6-13
 - PrinterTimeout 6-14
 - Quantity 6-14
 - TrimLeadingSpaces 6-15
 - WebAddress 6-15
 - WebPort 6-15
 - WebUserName 6-16
 - Internet Data Push
 - Configuring Components 3-24
 - Configuring Connections 3-27
 - Configuring Printers 3-31
 - Installing Components 3-24
 - Overview 3-22
 - Web Listener 3-35
 - Internet Printing
 - Data Push 3-22
 - Performance / Licensing Considerations 3-1
 - Web Client 3-2
- ## L
- Launch Browser 2-29
 - llmwcInt.ini 2-15
 - llmwcInt.ini file
 - Syntax 5-6
 - Loftware Mirror Manager 4-5
 - Loftware Print Server
 - Architecture 1-1
 - Configuration 1-12
 - Data Structures 1-22
 - Failover Support 4-4
 - Field Lists 1-30
 - Installation 1-7
 - Interactive Mode 1-21
 - Interfaces 1-33
 - Overview 1-1
 - Performance
 - Considerations 1-5
 - Service Mode 1-20
 - Troubleshooting 1-42
 - Loftware Virtual Server 4-15
 - Log Files
 - Event logs 1-17
 - Trace logs 1-16
 - LPSSend Client 2-30

M

Mirrored Registry 4-5

N

Notification Agent
Configuring 1-18, 2-3
Overview 2-2, 2-9
Starting 2-7
Troubleshooting 2-9

O

On-Demand Print Client
2-13
llmwcInt.ini 2-15
Operation 2-16
Using 2-14

P

Palm OS SDK 2-30

Q

Quorum Resource 4-3, 4-8

R

RAID 4-3
Remote Administration of
a Cluster Server 4-18
RFID
EPC Data Passback 1-37

S

Sales and Customer
Service 1-2
Shared Files 4-5
Status Client
Examples 2-24
Menu Choices 2-26
Operating 2-22
Overview 2-21
Screen Layout 2-22

Using Launch Browser 2-29

T

Technical Support I-3
Thin Client
Creating an Icon to Run
Client 2-19
Installation 2-2
LPSSend Client 2-30
Notification Agent 2-2
On-Demand Print Clients
2-13
Overview 2-1
Palm OS SDK 2-30
Sample Palm SDK
Program 2-31
Status Client Application
2-21
Troubleshooting 2-20
Using with a Database 2-17
vs On-Demand Print 2-14
Troubleshooting
LPS Clustering 4-20
On-Demand Print Client 2-20
Web Client 3-20

U

Uninstalling 4-18
Upgrading 4-18

V

Virtual Server 4-3, 4-15

W

Web Client
Components 3-6
Installing Components 3-8
Other Information 3-20
Overview 3-2
Using 3-18
vs. On-Demand Print 3-4

