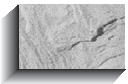




Deploying Large-Scale Labeling Systems



Creating large-scale label printing systems is never an easy task. However, with the proper amount of forethought and clever system design, large-scale printing systems can be achieved with the Loftware Print Server (LPS). This chapter outlines some important things to consider when your system is large and complex. It shows how to build and use a database of information pertaining to multiple LPS servers, their printers, and label formats. Your application can use this information for a more efficient and reliable system. The following discussion is a sample based on a mixture of Loftware customers. Your system will be different from the scenario given here, but most of the principles still apply.



NOTE

The purpose of this chapter is to highlight the issues in integrating Oracle Applications and Loftware in a holistic manner. Some of the setup steps outlined in this appendix are already available in Oracle Applications and are covered in detail in Chapter 15 of the book.

Scenario

This scenario is similar to an existing Loftware customer. In this chapter, they are referred to as Acme Widgets. The following sections summarize their topography and requirements.

ERP Application Deployment

Acme has 100 different plants throughout North America. Each plant has 10–15 barcode label printers, some have up to 50 printers. Their Oracle Applications installation is UNIX-based and will make label requests to the LPS.

Each plant has several workstations connected to Oracle. Users will use label-configuration features in this application to make label requests. Oracle will process these requests and pass them to the LPS.

Loftware Print Servers

In this case, because they wish to centralize all LPSs, and they have upwards of 1,500 printers, they will put servers running LPS at their data center. To maximize throughput, each LPS server will drive 200+ printers, making a total of seven servers. The alternative to this would be to regionalize the system and place an LPS server at each plant (or group of local plants).

An exception is that one of their plants has 50 printers, the WAN connection to this plant has limited bandwidth. In order to minimize bandwidth use and maximize performance at this location, an LPS server will be installed locally. The same applications that drive the centralized LPS servers will also drive the localized LPS servers.

The seven servers will be on Windows NT/2000 connected to the printers at the various plants across the company WAN.

To avoid the need for cross-platform connectivity software between UNIX and NT/2000 (like Samba), the Oracle Application will interface with the LPSs via a TCP/IP socket.

Printers

Over the years, Acme Widgets has purchased several different barcode printers from various manufacturers. They do not want to terminate these printers in favor of purchasing one type of printer for all plants. This makes the system more complex, but can be handled if the system is well organized.

They have Intermec 3400s, Intermec 3440s, Printronix 5206s, Zebra 170xi2s, and Zebra 170xi3s. Each printer is connected to the company WAN and has its own unique TCP/IP address.

Each printer is on a maintenance plan where the printhead, sensors, and mechanisms are cleaned on a regular basis. This reduces label-printing errors, which are difficult to troubleshoot (and a very common problem).

Label Formats

Acme Widgets has more than 70 different customers that require their shipments to be barcoded. New customers are being added every day. Each customer has different packaging levels, which require different label formats. Some customers require up to 10 different types of labels.

Many of the labels require complex 2D symbologies such as PDF417, microPDF, and Maxicode. More than 500 different label formats will be needed. These formats will be stored in a centralized location on a special RAID storage drive. Special provisions will be made so that Oracle cannot send a label format to a printer for which it was not designed (a very common problem).

The following discussion demonstrates a scenario for building a database for keeping track of the servers, printers, workstations, and label formats. Using these tables, a general request coming from a workstation can be resolved into a specific print request for the LPS. The details of your system will be different, but the general concepts apply. It is extra work to track all this information in your application, but the consequences of not doing so can be severe—very upset users with all kinds of printing problems.

Making Label Requests

After the initial setup of the LPS servers, printers, and label formats, making a label request to the LPS is relatively easy. Three different formats can be used for a request: PAS, CSV, and XML. These formats are discussed in the Loftware manual or online guides. Each format contains the same data, just in a different syntax. This example will use the PAS syntax for clarity. To print a Ford mixed load label on printer 5 for a red widget weighing 10 lbs, the request would look like this:



```
*FORMAT,\\labelServer\labelMasters\zebra\ford_mixed_z170xi2.lwl
*PRINTERNUMBER,5
COLOR,RED
DESCRIPTION,WIDGET
WEIGHT,10 lbs.
*PRINTLABEL
```

This information would then be sent to the designated LPS server via a file drop or through a TCP/IP socket.



NOTE

This example is simplistic in nature to give you some insight into what a request generally looks like. Your requests and how you derive them will obviously be different.

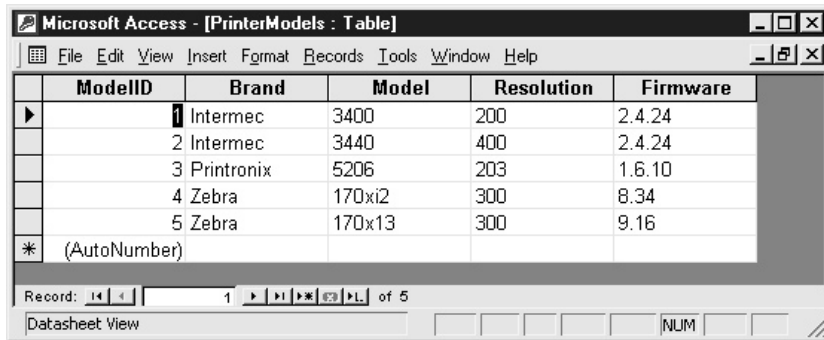
Organizing Printer Types

Because most companies have a mixture of printers from different manufacturers, it becomes helpful to organize them into a database. Any program acting as a front-end to the LPS can use a table similar to this. The main use for this table is to make sure that requested label formats are targeted to the printer for which they were designed. Of course, it can also be used and modified as the designer sees fit. Figure 1 shows the printer models table.

Model ID is the table key. Brand specifies the Printer Manufacturer. Model represents the actual Model Number of Printer. Resolution indicates the printer resolution in DPI. Firmware identifies the firmware version of the printer. The fields Resolution and Firmware are optional.

Organizing Printers

The next step is to create a table that lists all of the printers in your system as shown in Figure 2. This table will refer to the Printer Model Table as well as to the table of LPS Servers, which is discussed in the next section.

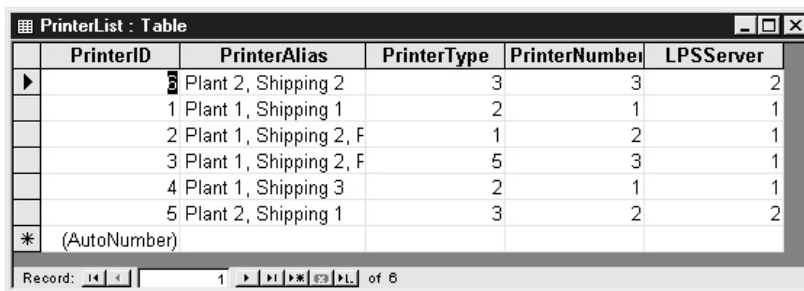


	ModelID	Brand	Model	Resolution	Firmware
▶	1	Intermec	3400	200	2.4.24
	2	Intermec	3440	400	2.4.24
	3	Printronix	5206	203	1.6.10
	4	Zebra	170xi2	300	8.34
	5	Zebra	170x13	300	9.16
*	(AutoNumber)				

Record: 1 of 5
Datasheet View

FIGURE 1. *Printer Models Table*

PrinterID is the Table key. PrinterAlias is the “human readable” name of the printer. This can match the alias given to the printer when configuring printers in label design mode. If this is the case, the alias can be used instead of the printer number when making a request for a particular printer. The PrinterType references the Printer Model Table. PrinterNumber is the number that the printer is associated with in the corresponding LPS printer list. LPSServer is the server number where the printer can be found. This is a foreign key into the LPSServers table.



	PrinterID	PrinterAlias	PrinterType	PrinterNumber	LPSServer
▶	3	Plant 2, Shipping 2	3	3	2
	1	Plant 1, Shipping 1	2	1	1
	2	Plant 1, Shipping 2, F	1	2	1
	3	Plant 1, Shipping 2, F	5	3	1
	4	Plant 1, Shipping 3	2	1	1
	5	Plant 2, Shipping 1	3	2	2
*	(AutoNumber)				

Record: 1 of 6

FIGURE 2. *Printer List Table*

Organizing LPS Servers

If you are using multiple LPS servers, it becomes necessary to keep track of them. Your application will have to know how to contact the server for the purposes of making a label request. Again, the best way to do this is with a database table, as shown in Figure 3.

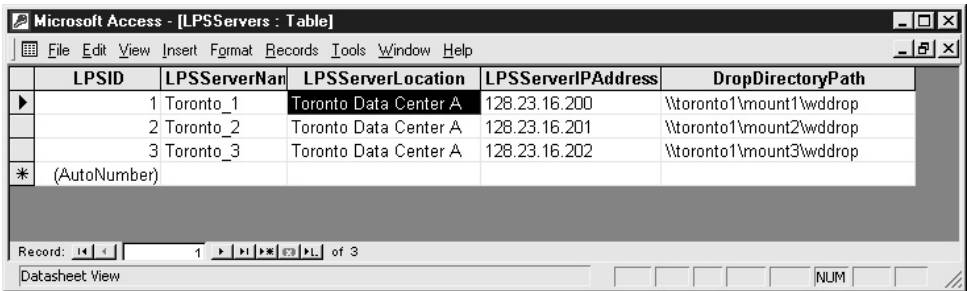
LPSID is the Table key. LPSServerName is the name of LPS Server. This is for reference only, and you can use any naming convention that you choose. The LPSServerLocation identifies the physical location of LPS server. Again, this is for reference only and you can use any naming convention. The LPSServerIPAddress is the TCP/IP address of the LPS server. This is for the purposes of making print requests via a socket as opposed to a file drop. The DropDirectoryPath is the path to the directory that the particular LPS is scanning. This is for the purpose of making print requests to the LPS via a file drop.

Organizing Label Formats

Each printer manufacturer has its own proprietary language, and each model of that family will have different print resolutions and capabilities. Unfortunately, these differences force you to design a different label for each type of printer you have. This can quickly become cumbersome if not properly organized.

The first step to having a well-organized system is to establish a procedure for tracking label formats. The best way to do this is with a database. Figure 4 shows a database table that can be used for tracking your label formats along with their corresponding printers. Notice that the PrinterType field is a foreign key to the table in Figure 1. This is how the front-end application will know what label is designed for what printer.

LabelIndex is the Table key. The LabelDesignation is a general way to specify a label format without having to know the exact name of the format that will be needed for the specified target printer. Notice that there are two FML entries (Ford Mixed Load). One entry is for printer type 4, the other for printer type 5. It will be



	LPSID	LPSServerName	LPSServerLocation	LPSServerIPAddress	DropDirectoryPath
▶	1	Toronto_1	Toronto Data Center A	128.23.16.200	\\toronto1\mount1\wddrop
	2	Toronto_2	Toronto Data Center A	128.23.16.201	\\toronto1\mount2\wddrop
	3	Toronto_3	Toronto Data Center A	128.23.16.202	\\toronto1\mount3\wddrop
*	(AutoNumber)				

Record: 1 of 3
Datasheet View

FIGURE 3. LPS Servers Table

LabelIndex	LabelDesignat	LabelName	LabelPath	PrinterType	LabelDescription
1	GMS	GM_STANDARD_I3440	WabelServer\LabelMasters\intermec\3440	2	GM Standard, I3440
2	GMM	GM_MASTER_I3440	WabelServer\LabelMasters\intermec\3440	2	GM Master, I3440
3	GMML	GM_MIXED_I3440	WabelServer\LabelMasters\intermec\3440	2	GM Mixed, I3440
4	GMS	GM_STANDARD_P5206	WabelServer\LabelMasters\pintronix\5206	3	GM Standard, P5206
5	GMM	GM_MASTER_P5206	WabelServer\LabelMasters\pintronix\5206	3	GM Master, P5206
6	GMML	GM_MIXED_P5206	WabelServer\LabelMasters\pintronix\5206	3	GM Mixed, P5206
7	FS	FORD_STANDARD_Z170xi2	WabelServer\LabelMasters\zebra\170xi2	4	FORD Standard, Z170xi2
8	FM	FORD_MASTER_Z170xi2	WabelServer\LabelMasters\zebra\170xi2	4	FORD Master, Z170xi2
9	FML	FORD_MIXED_Z170xi2	WabelServer\LabelMasters\zebra\170xi2	4	FORD Mixed, Z170xi2
10	FS	FORD_STANDARD_Z170xi3	WabelServer\LabelMasters\zebra\170xi3	5	FORD Standard, Z170xi3
11	FM	FORD_MASTER_Z170xi3	WabelServer\LabelMasters\zebra\170xi3	5	FORD Master, Z170xi3
12	FML	FORD_MIXED_Z170xi3	WabelServer\LabelMasters\zebra\170xi3	5	FORD Mixed, Z170xi3

FIGURE 4. *Label Format Table*

demonstrated later how a workstation can make a request for an FML label, and the Oracle program will be able to derive the format name, printer number, and server address.

The LabelName is a unique name of label format as given in the Loftware Label Designer. The LabelPath is the path to label. In this example, formats are organized in a different folder for each printer model. The PrinterType is a reference to the printer table for the printer model that the format was designed for. The LabelDescription describes the label and is for reference purposes only.

Tracking Work Stations

Depending on the nature of your system, you might have many workstations making requests. If this is the case, the table shown in Figure 5 suggests how you could organize these workstations in a way that fits into your schema for formats and printers.

StationID	StationAlias	StationDesignator
1	Plant 1, Shipping 1	P1S1
2	Plant 1, Shipping 2	P1S2
3	Plant 1, Shipping 3	P1S3
4	Plant 2, Shipping 1	P2S1
5	Plant 2, Shipping 2	P2S2

FIGURE 5. *Stations Table*

NOTE

Two tables are needed because a workstation can have multiple printers.

StationID is the Table key. StationAlias is a “human readable” description of this station. StationDesignator is the designator that is contained in a request coming from this workstation. Figure 6 shows the station printers table.

StationPrinterID is the Table key. PrinterID is the foreign key into printers table for obtaining printer information. StationID is the station that the printer belongs to. In this case, stations may have more than one printer. StationPrinter is the designator for the target printer contained in a request coming from the workstation.

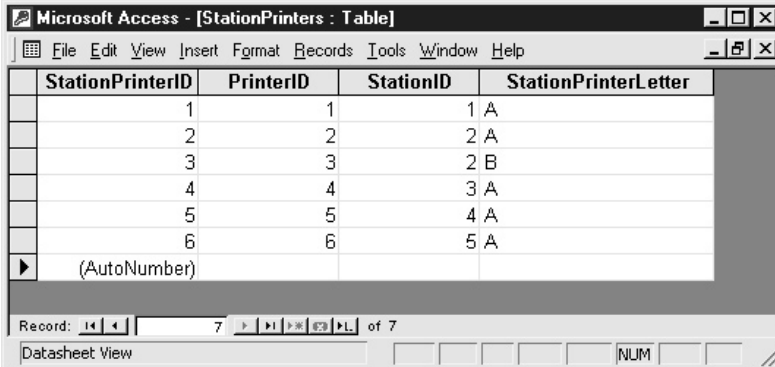
Table Relationships

All the various tables discussed until now are related to each other through foreign key relationships. The relationships between the various tables in the architecture are shown in Figure 7.

How This Database Scheme Is Used

Okay, now that we have defined all these tables and relationships, how are they used by the application? The best way to describe this is with an example of a transaction coming from a workstation.

As stated in the scenario given at the beginning of this chapter, requests are made from a workstation at one of the plants. Oracle is prompting the user for the following:



StationPrinterID	PrinterID	StationID	StationPrinterLetter
1	1	1	A
2	2	2	A
3	3	2	B
4	4	3	A
5	5	4	A
6	6	5	A
(AutoNumber)			

FIGURE 6. Stations Printers Table

- A key field into the database
- The label format designation (used for later lookup in the Label Formats table)
- The printer that they want to print on (workstations can have more than one printer)

Figure 8 shows a pseudo string that is generated by Oracle Applications based on the workstation. The operator is not prompted for the station code; the program is smart enough to know which code is associated with it and includes it in the return string. This code will be used to look up the printer types and addresses associated with the workstation.

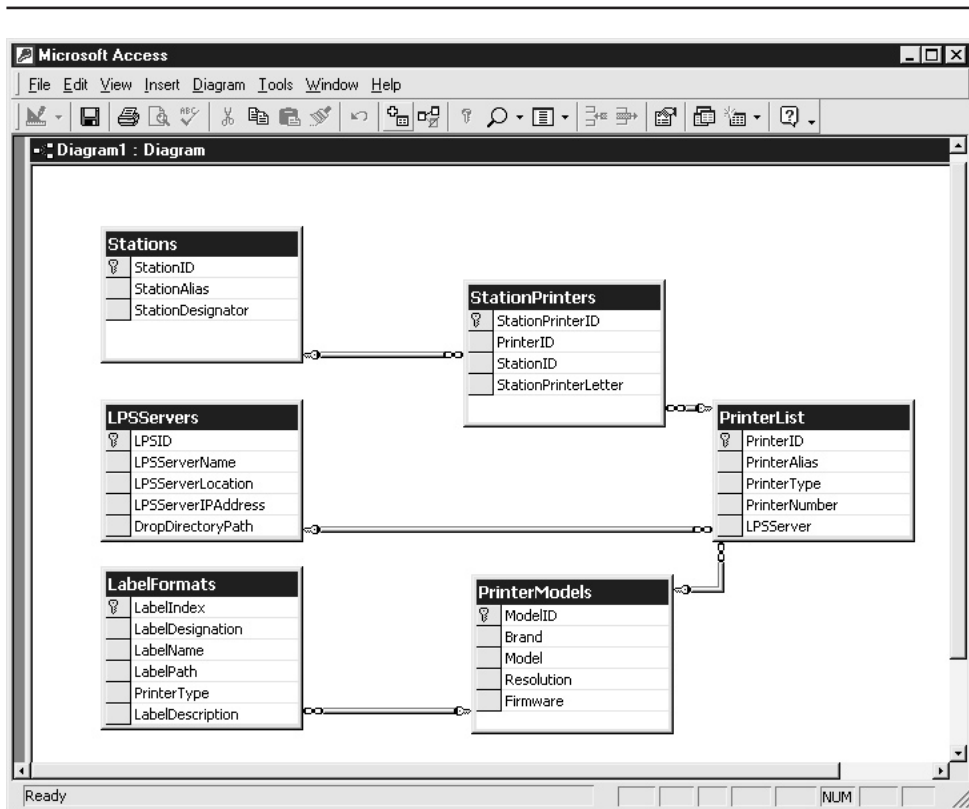


FIGURE 7. *Table relationships*

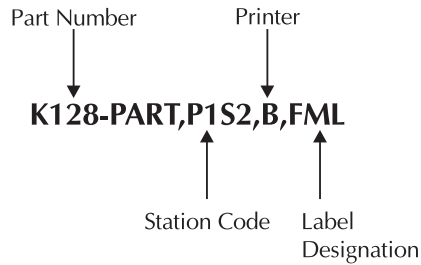


FIGURE 8. *Pseudo return string built by Oracle Applications based on the workstation*

NOTE

This is a top-level example, it is not meant to resemble what the actual codes sent back from the client request will look like.

Notice that P1S2, B, and FML are all in the sample tables that were presented earlier. At this point, the Oracle Applications take this information, uses the database to get the rest of the information and generate the label request for the LPS server. The following steps illustrate this:

1. Retrieve all Acme Widgets–specific information from the company database based on the key field retrieved from the workstation request. In Figure 8, the key is K128-PART. (Your key could be anything.)
2. Use SQL Select statements using the station code (P1S2) and printer code (B) to obtain the printer number, printer type, and LPS server information for this request. The printer alias could also be retrieved if you are using aliases instead of numbers. The following SQL statement does this using the station code and printer data:

```

SELECT PrinterList.PrinterType, PrinterList.PrinterNumber,
LPSServers.LPSServerIPAddress,
LPSServers.DropDirectoryPath
FROM PrinterList INNER JOIN
StationPrinters ON
PrinterList.PrinterID = StationPrinters.PrinterID INNER JOIN

```

```
Stations ON
StationPrinters.StationID = Stations.StationID INNER JOIN
LPSServers ON
PrinterList.LPSServer = LPSServers.LPSID
WHERE (Stations.StationDesignator = 'P1S2') AND
StationPrinters.StationPrinterLetter = 'B')
```

Results from this query show that the printer type is 1, the IP address of the LPS server is 128.23.16.200, the drop directory path is \\toronto1\mount1\wddrop, and the printer number is 5. Manually compare this result to the tables, and you will see that it is correct, as shown in Figure 9.



NOTE
Because Acme Widgets is connecting to the LPS through a socket, all it needs is the server IP address. The drop directory path is included for applications that want to make requests via a file drop.

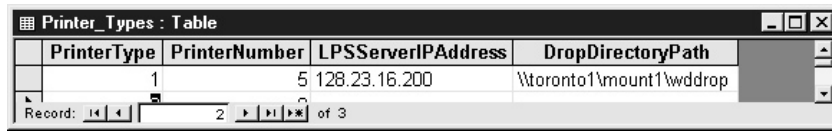
- 3. Use a **select** statement to obtain the format name and path corresponding to the label designator and printer type. This ensures that the label specified in your request to the LPS is designed for the target printer. Using this technique also preempts the need to know the exact name of the label format ahead of time.

```
SELECT LabelPath, LabelName
FROM LabelFormats
WHERE (LabelDesignation = 'FML') AND (PrinterType = '5')
```

Figure 10 shows the results of this query. These results can be combined to get the full path name of the label format for the *FORMAT LPS command.

Label_Path : Table	
LabelPath	LabelName
\\LabelServer\LabelMasters\zebra\170xi3	Ford_Mixed_Z170XI3

FIGURE 9. *The printer type, printer number, LPS IP address, and drop directory path are all obtained*



PrinterType	PrinterNumber	LPSServerIPAddress	DropDirectoryPath
1	5	128.23.16.200	\\toronto1\mount1\wddrop

Record: 1 of 3

FIGURE 10. *The label path and label name are selected***NOTE**

Had this query returned a NULL result, it would indicate that there is no label format of type “FML” designed for printer type 5”. Displaying an error message at this point would be paramount to solid system design.

4. Use the information obtained in the first three steps to generate your request to the target LPS server. Your request will look like this if you are using pass files:

```
*FORMAT,\\labelServer\labelMasters\zebra\170xi3\
ford_mixed_z170xi3
*PRINTERNUMBER,5
COLOR,RED
DESCRIPTION,WIDGET
WEIGHT,10 lbs.
*PRINTLABEL
```

5. Open a TCP/IP connection to the target LPS server, which in this case is 128.23.16.200 and send your pass file through the socket. You can use one of two modes for your request:

- Wait mode means that your code will wait until it receives an acknowledgement back from the LPS that the label either printed, or an error occurred. This allows you to take action if an error occurs. The problem with this is that if your label is “in queue,” you will be in a wait state until it prints in systems with high throughput requirements—this is not optimal.

- No Wait mode means that you send your job, the LPS responds immediately that it has received the request, and you begin sending your next job. This is the fastest way to go, but you are running “open loop”—you do not have confirmation that the label actually printed. Some companies use the Loftware Notification agent to partially fill this gap.

Other Important Large-Scale Topics

There are several other topics to consider when using the LPS in a large system that are beyond the scope of this discussion. You can find information on these topics in the Loftware Advanced Users Guide and on the Loftware Web site (www.loftware.com) in the form of white papers, manuals, and online guides:

- Selecting printers that are up to the task
- Label design strategies for maximum throughput
- TCP/IP LPS interface for avoiding the need to mount drives between different operating systems, providing printer status to your application, and maximizing speed
- Analyzing your loads and strategizing how many servers you will need
- Strategies for localizing and centralizing the LPS servers on your WAN and Internet
- Understanding Internet functionality and when to use it

As you build your system, you will come up with new ideas that you could incorporate into your database to make it easier to maintain and more reliable. It is our experience that the more time put into designing your front-end application, the faster and more reliable it will be.