Leica LP C

(Laser Printer for Cassettes)

System Requirement & Specifications

Copyright© 2010 Leica Microsystems All rights reserved

Scope

This document contains basic information on the requirements, specification and connectivity options for the LP C Software System. This manual does **not** provide instructions for the installation or upgrade of hardware.

Disclaimers

This manual is not a substitute for the detailed operator training provided by Leica, or for other advanced instruction. A Leica representative should be contacted immediately for assistance in the event of any instrument malfunction.

Table of Contents | LP C Software System

Section 1 | Introduction

TheLabeLase software system allows a laboratory to utilize bar code technology for positive identification and tracking of specimens in the laboratory. The use of bar code technology results in less errors, greatly enhanced workflow efficiencies, and laboratory technicians that spend less time worrying about specimen identification errors and more time on diagnostics, results and patient care.

TheLabeLase software system may be implemented as a stand-alone system or integrated with an information system.

Section 2 | Software System Components

Label Printer Drivers

Leica provides the drivers and settings for the label printers that can be installed on supported Microsoft Windows® operating systems. The drivers may be installed on the computer system connected to the device directly or a server if utilizing a network connection.

LabeLase® Producer CL

LabeLase® Producer CL is the end user interface that allows the computer system to communicate to the attached laser cassette marker. LabeLase® Producer CL provides a user access to the LP C print queue, status and settings. Its use in a laboratory will depend on the level of integration. Users have the option of entering data for printing into LabeLase® Producer CL or if interfaced into the Lab Information System or other application, LabeLase® Producer CL can be minimized or run in the background. This software must be running to allow the LP C to print.

Section 3 | System Requirements

LabeLase® Producer CL Requirements

Processor	Pentium 400MHz or higher
Memory	128 megabyte (MB) RAM or higher
Hard Disk	50 megabyte (MB)
Display	800x600 or higher resolution monitor
Operating System	Microsoft Windows® 2000, Microsoft Windows® XP, Microsoft Windows® Vista or later operating system
Network	100 megabits per second (Mbps) TCP/IP connection or higher to the same subnet the printer will be installed on. (required if marker will not use a serial connection)
Ports	An available DB9 serial or USB port (required if the LP C will not be on a network)

Section 4 | **LP C** Interface Specifications

Standard Message Format

The Standard Message Format consists of several fields of information, each separated from the others by a separator character (or sequence of characters) specified in the Data Field Separator of the LabeLase® Host Communications form. Each set of records within the message needs to be terminated by carriage return (CR) and line feed (LF) characters. A sample message format with is defined below using the comma character as the separator:

<FILENAME>,<QUANTITY>,<COPIES>,<SERIALNO>,<VMAGID>,<EXITBIN>,<FIELD1>,<FIELD2>, ...<FIELDN>[cr][lf]

Field Name	Description	
Required	Default	Example
Layout Filename	Name of the file that contains the layout to be printed. This may optionally include the entire path. The Producer software will load this file prior to printing. This allows the lab software to dynamically change the layout of the data on the cassette.	
No	N/A	C:\Program Files\GDC\Template\Sample.itl
Quantity	Defined the number of cassettes to print.	
No	1	3
Copies	Reserved, no value should be sent.	
No	1	N/A
SerialNo	Reserved, no value should be sent	
No	N/A	N/A
VMAGID	Three digit numeric value which represents the cassette color	
12 marker only	None	101
Exit Bin ID	Bin where printed cassette is printed. Maybe be defined as “1”, “2”, “3” or “any” or “*” (do not include the double quote marks.)	
12 marker only	None	Any
Data Field <…>	Data fields from an application to be printed and/or encoded in the barcode.	
Yes	None	S09-26743

Table 1 - Message Fields Defined

Preferred Message Format

Leica prefers that the following message format is used using a comma as a delimiter.

```
<FILENAME>,<QUANTITY>,<VMAGID><EXITBIN>,<FIELD1>,<FIELD2>, ...<FIELDN>[cr][lf]
```

This allows the greatest flexibility in allowing the system to use a single or twelve laser cassette marker. When printing to a single laser cassette marker (LP C) and using this format send any 3 digit number (101 for example) as the VMAGID.

Below is an example of five records coming over using this message format:

```
“C:\Program Files\LPC\Template\Sample.itl”,1,101,any,S11-1234,A,1[cr][lf]
“C:\Program Files\LPC\Template\Sample.itl”,1,101,any,S11-1234,A,2[cr][lf]
“C:\Program Files\LPC\Template\Sample.itl”,1,101,any,S11-1234,B,1[cr][lf]
“C:\Program Files\LPC\Template\Sample.itl”,1,101,any,S11-1235,A,1[cr][lf]
“C:\Program Files\LPC\Template\Sample.itl”,1,101,any,S11-1236,A,1[cr][lf]
```

Connection Types

Below is an overview of the available connection types.

RS232-C Serial COM Port

The message is captured via a serial port connected to the computer system running the LabeLase® Producer CL software.

Network TCP/IP

The message is sent to a user defined TCP/IP port on the IP address of the computer system running the LabeLase® Producer CL software.

File Transfer

The message is written to an ASCII text file and placed in a folder that is configured in the LabeLase® software. The extension and location of the file are configurable.

InfoSight® Extended Protocol

The InfoSight® Extended protocol allows bidirectional communication between an information system and the LabeLase® software utilizing any of the connection types described above. The information system acts as the master in the master-slave relationship. The LabeLase® Producer CL will not send any messages to the master unless it is requested. Details for utilizing this protocol are below.

Protocol Parameters

Communications is established via an RS232 serial connection with the following parameters:

Communications Parameters

Baud Rate	Selectable 1200 – 19200
Data Bits	8
Parity	None
Start Bits	1
Stop Bits	1
Three Wire	RX, TX and GND

Software handshaking (XON/XOFF) is implemented in some types of marking systems to control the flow of information. Hardware handshaking (DTR/DSR) is not implemented.

Communications with the marking system controller is accomplished via a Master / Slave arrangement with the Host being the Master and the marker being the slave. The slave will only transmit in response to a message from the master.

Note: All transmissions are in standard ASCII utilizing the following control characters:

Control Character Definitions

Character	Definition	Hex	Decimal
SOH	Start of Header	0x01	1
STX	Start of Text	0x02	2
ETX	End of Text	0x03	3
CR	Carriage Return	0x0D	13
ACK	Acknowledge	0x06	6
NAK	Negative ACK	0x15	21
XOFF	Transmit Off	0x13	19
XON	Transmit On	0x11	17

Master Data Format

SOH TYPE STX [DATA TEXT] ETX [BCC] CR

Where,
TYPE - A single printable ASCII character that defines the meaning and the contents of the message [DATA TEXT] field. Message types may be custom defined for certain applications as required. Standard message types are defined later.

[DATA TEXT] - An optional field which contains the actual data of the transmission. Some message types require no data since the "message" is conveyed by the TYPE character.

BCC - This is an optional field used to improve link reliability by providing fault detection. The BCC is computed by taking an eight bit addition of the TYPE and DATA TEXT characters and transmitting them as a three digit ASCII decimal number in the range 000 to 255. Refer to the example BCC computation later in this document.

Slave Data Format (response)
The slave will respond to the master's transmission in one of two ways depending on whether errors were detected or not.

```
    SOH   TYPE   ACK   STX   [DATA TEXT]   ETX   BCC   CR
or
    SOH   TYPE   NAK   STX   [DATA TEXT]   ETX   BCC   CR
```

If no errors were detected in the reception of the packet, then the first response will be sent back to the master. If any errors were detected (e.g. Parity, Framing, Overrun, BCC, Format, etc.) then the second (NAK) message will be sent. Note that the ACK message does not necessarily imply that the DATA TEXT field itself is correctly presented, just that no communications errors occurred.

The TYPE character will always be the same as the received TYPE.

The DATA TEXT field is optional and depends on the message TYPE. The BCC field will always be present in the response.

Retries
If the host does not receive a response from the I-Dent within three seconds, or it receives a NAK response, it should retransmit the entire packet. If, after three retries (four tries total), the host has not received a response, the host should declare the link to be "down".

Standard Message Types
Certain message types which are common to most marking systems are briefly discussed below. For a complete description of the Extended Protocol message types for a particular marker, refer to the data sheet for that marker.

1 - The DATA TEXT is a message to be printed. The message text is stored in the buffer for the currently assigned buffer. The response message contains no data in the DATA TEXT field.

A - The DATA TEXT is a number from 1 to 10 specifying the buffer number to Assign for printing. The response message DATA TEXT field will contain an ASCII '1' if the assignment number is valid, or '0' if it is invalid.

S - This type is used to query the marker for its current status. There is no DATA TEXT field for this type. The ACK response message from the marker will contain a DATA TEXT field. This field will contain one or more numbers separated by commas where each number is made up of four ASCII digits representing a decimal value. The binary equivalent value of this number has a bit pattern representing the status of the marker and is dependent upon the specific marker and application.

Example BCC Computation
The following example is a typical transmission including the BCC.

To download the character string 'ABC123' to the currently assigned message buffer, send the following message.

```
SOH   1   STX   ABC123   ETX   141   CR
```

where '1' is the message TYPE and 141 is the BCC. The BCC is computed as follows (note all math shown in hexadecimal):

1) BCC = Message TYPE character + DATA TEXT characters.

```
          031H          1        - Message Type
          041H          A        \
          042H          B        |
          043H          C        | message Text
          031H          1        |
          032H          2        |
+         033H          3        /
______________
          18DH
```

2) We are only interested in the lower eight bits of the sum, so we discard the first digit and keep the lower two. This results in a BCC of 8DH. Note that when performing the summation in an eight bit variable (e.g. unsigned char in 'C') that the most significant bits are automatically truncated. If the master's programming language is incapable of doing eight bit addition, then the same result can be obtained by taking the MODULO 256 operation on a sixteen bit sum. The MODULO operation is division where the Remainder is kept and the Quotient is discarded.

3) Once the BCC value is obtained, it must be placed into the message packet after the ETX character. The BCC must be transmitted in its decimal ASCII form. The decimal equivalent of the hexadecimal value 8DH is 141 decimal. Converting the value 141 into three ASCII characters yields 031H, 034H and 031H. These three characters become the transmitted BCC.

The actual data transmitted (in hex) by the host for this message is:

```
001H   031H   002H   041H   042H   043H   031H
032H   033H   003H   031H   034H   031H   00DH
```

4) If the marker receives the message correctly, it will respond with the following message:

```
001H   031H   006H   002H   003H   030H   034H   039H   00DH
```

which equates to the following ASCII message:

```
SOH   1   ACK   STX   ETX   049   CR
```

Note that if the marker detected an error in receiving the message from the host it would respond with a NAK (015H) character in place of the ACK. The BCC would be unaffected by this since the ACK/NAK is not included in the BCC computation.