

Application Note

LABEL PRINTING INSTRUCTIONS

TABLE OF CONTENTS

<i>Assumptions.....</i>	<i>3</i>
<i>Requirements.....</i>	<i>4</i>
<i>Solution Overview</i>	<i>4</i>
<i>CALTools Software Setup.....</i>	<i>5</i>
Install CALTools	5
Setup CALTools	5
Install Printer Driver.....	5
Select Printer in CALTools	6
Set Working Directory in CALTools	6
Upgrading from DOS Version of CALTools.....	6
<i>Working with Label Templates</i>	<i>7</i>
Modifying Label Templates.....	7
Creating a file to be used by PASS	8
Using Label templates in PASS 6.0.....	9
Copying label template files and PASS program files from one computer to another.....	10
<i>Printing Variable Data.....</i>	<i>11</i>
Variable Data and Format Symbols	11
Variable Data and PASS 6.0 Messages Table	14
Variable Data and PASS 6.0 Messages: Justification and Padding	15
Printing Frequently Used Parameters.....	17
Printing Frequently Used Parameters.....	18
Printing Serial Numbers	18
Printing the Unit ID.....	23
Printing Operator Identification	24
Printing Scanned Bar Code Information	27
<i>Printing Labels Containing Graphics.....</i>	<i>29</i>
Adding a graphics object to a label format.....	29
Downloading a label containing graphics to the printer	30
Troubleshooting and managing printer graphics files	31
<i>Troubleshooting Common Label Printing Problems</i>	<i>32</i>

Label Printing Instructions

This document explains how to use the PASS 6.0 software and Create-A-Label Tools software to program the Dynalab Analyzer to print labels.

This document contains the following main sections:

- 1 a list of assumptions – knowledge required to perform the tasks outlined in this document
- 2 a list of required hardware and software
- 3 an overview of the Dynalab solution to the problem
- 4 instructions regarding installation, setup, and upgrading
- 5 an explanation of how to work with label templates
- 6 an explanation of how to generate parameters frequently used on labels such as serial numbers, Unit ID, Operator ID.
- 7 an explanation of how to print scanned barcode information
- 8 an explanation of how to print labels which contain graphics
- 9 an guide to troubleshooting common label printing problems

Assumptions

To successfully use this document, the following knowledge is required:

- basic knowledge of how to enter data in the messages table using PASS 6.0
- knowledge of how to use the Sequence table to create a Sequence

For assistance on how to use features of PASS 6.0, see the PASS 6.0 Help file.

Requirements

To successfully print labels using a Dynalab Analyzer, the following hardware and software are needed:

Hardware:

- Dynalab Circuit Analyzer
- Dynalab Label Printer
- Printer Cable 7430001 for serial communication between the Analyzer and printer or computer and printer. This cable connects the serial port of the printer to serial port 2 of the Analyzer.

Software:

- PASS Version 6.00 or higher
- Create-A-Label Tools, Version 3.04

Solution Overview

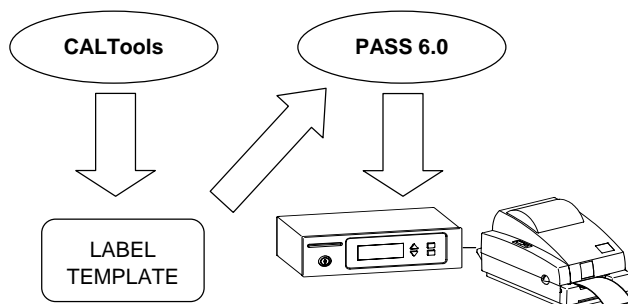
Programming the Analyzer to print labels involves the following steps:

- 1 Create a label template, using Create-A-Label Tools software
- 2 Generate a label template source file to be used by PASS software
- 3 Include a label printing Sequence item in the PASS program which references a label template as defined in steps 1 and 2

Create-A-Label Tools (CALTools) is a separate software package, which is supplied on the same installation CD as PASS 6.0.

To launch CALTools from within PASS 6.0, click on the **Labels** button on the main toolbar.

CALTools software is used to create label templates. A label template specifies what information is printed on a label, where the information is printed on the label, and what the size of the label is. A label template usually contains variable information.

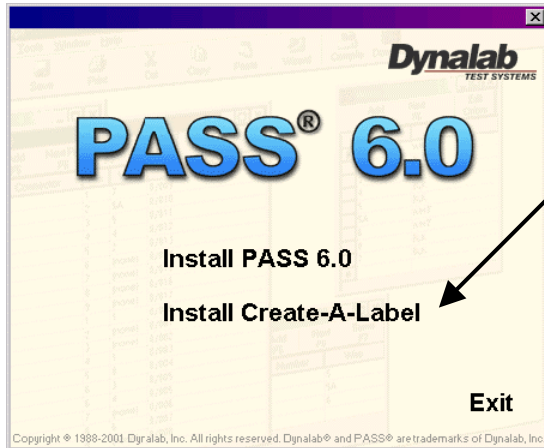


The label template is incorporated into the PASS program and becomes part of the compiled executable that is downloaded to the Analyzer. When printed, the variable information specified in the label template is filled in with specific data.

CALTools Software Setup

Install CALTools

Create-A-Label Tools (CALTools) software is provided on the PASS 6.0 Installation CD. To install CALTools, insert the PASS 6.0 Installation CD in the computer's CD drive. The installation should begin automatically. If not, run the setup.exe file directly from the CD-ROM drive.



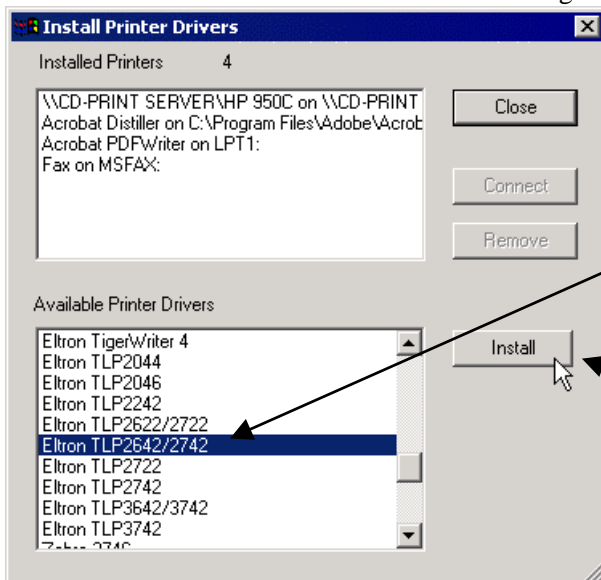
This is the first screen presented by the installer. It provides a menu of choices. To install CALTools, select the menu item that says "Install Create-A-Label"

During the installation process, several screens will be presented. Accept all the defaults presented by these screens. The last step of the installation will restart the computer.

Setup CALTools

Install Printer Driver

After the computer has restarted, start the CALTools software. The following screen will be presented:



In the "Available Printer Drivers" pane, scroll down and select ELTRON TLP2642/2742, then press the Install button.

1 – Highlight **Eltron TLP2642/2742**

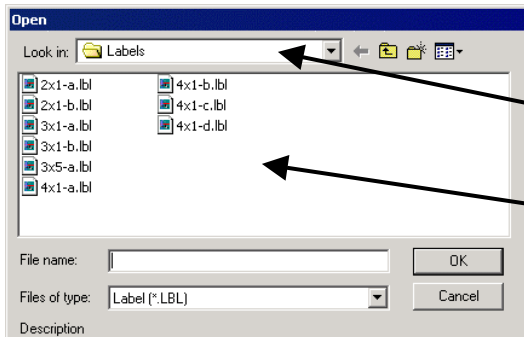
2 – Press **Install** button

Select Printer in CALTools

In CALTools, perform the following tasks:

1. From the **File** menu, choose **Select Printer...**
2. Select the printer that was installed above. Press **OK**

Set Working Directory in CALTools



In CALTools, perform the following tasks:

1. From the **File** menu, choose **Open**
2. From the Open window, select the directory to correspond to the default labels directory:
C:\DYNALAB DATA\LABELS
3. Select one of the predefined label templates in this directory and select **OK**.
4. CALTools will open the selected file. The working directory is now properly set. Exit CALTools.

Upgrading from DOS Version of CALTools

If a DOS version of CALTools was previously installed on the system, the label files created under the DOS version may be used with CALTools Version 3.04. The files must be copied from the directory in which they were stored by the DOS version to the new CALTools working directory.

The label files are stored by the DOS version in: **C:\PASS\CALTOOLS**.

The label files are stored by Version 3.04 in:

C:\DYNALAB DATA\LABELS.

Copy all files whose extensions are .lbl or .ejf

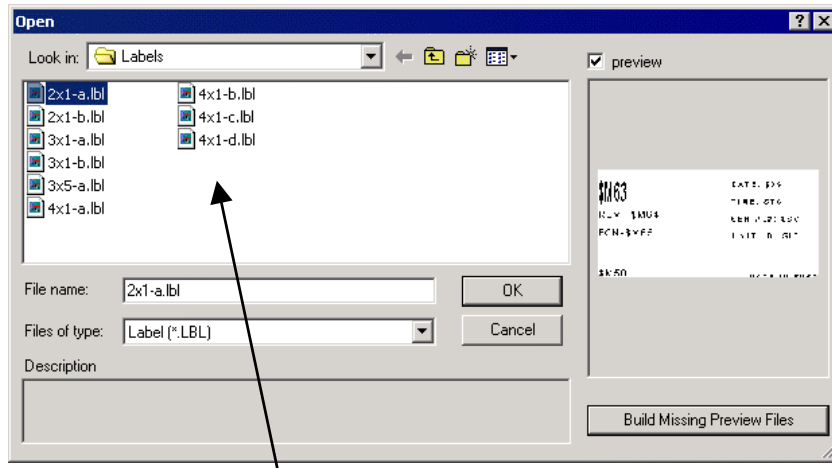
from:

C:\PASS\CALTOOLS

to

C:\DYNALAB DATA\LABELS

Working with Label Templates



Pre-Defined Label templates

Dynalab supplies nine pre-defined label templates.

Two templates are supplied for 2"x1" labels, two for 3"x1" labels, one for 3"x5" labels, and four for 4"x1" labels. The default location for these files is C:\Dynalab Data\Labels

In CALTools, to select a pre-defined label template, select File -> Open. A window similar to the one shown at left will appear.

Note: The recommended approach to designing a label template is to select one of the pre-defined formats, and modify it.

Modifying Label Templates

To modify a label template, the capabilities of CALTools to create objects (lines, boxes, bar codes, and text) should be used.

WARNING: When adding text, do not use True Type fonts – they are not supported by the Dynalab printer. Only use one of the fixed width fonts.

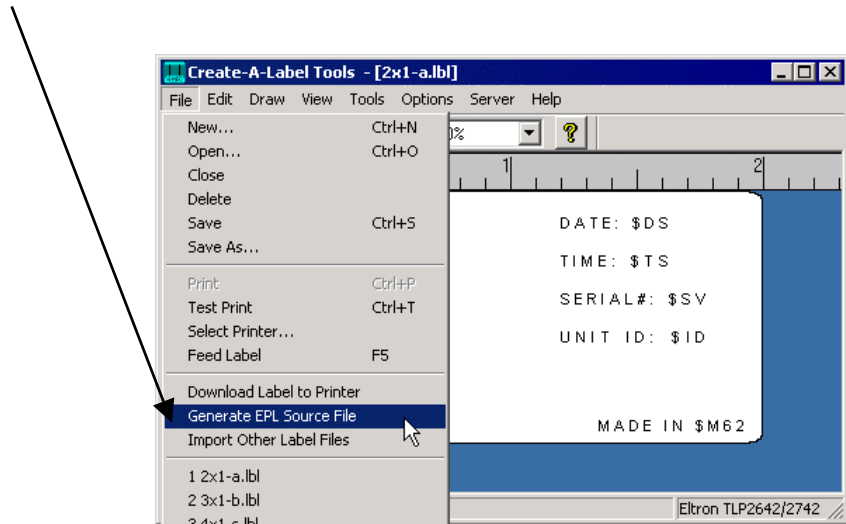
The following summarizes the steps required to open a pre-defined label template and make modifications to create a custom label template:

- 1 Click on the **Labels** button that appears on the toolbar in PASS 6.0.
- 2 After CALTools launches, select **Open** from the **File** menu.
- 3 Select the sample label from the list that matches the desired label size.
- 4 There are at least two sample labels for each common size of label.
- 5 Select **Save As...** from the **File** menu to save the label with a new name.
- 6 Adjust or Add content to the label as required. Refer to the CALTools Help file for instructions.
- 7 When finished with the modifications, the label template must be saved to the Labels folder as specified in PASS Settings. Usually, this is: **C:\Dynalab Data\Labels**. This file will have a .lbl extension.

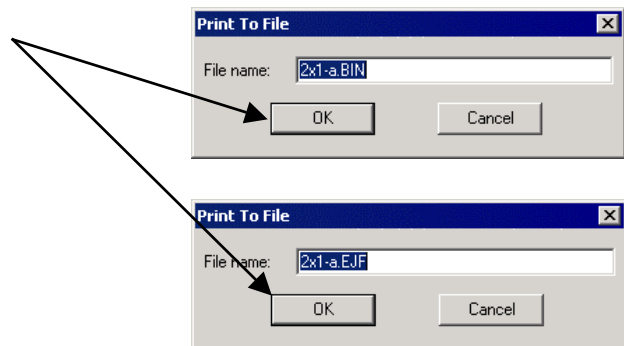
Creating a file to be used by PASS

After creating the label template file as described in the previous section, a source file that can be used by PASS 6.0 must be generated. This is done in CALTools by selecting:

File -> Generate EPL Source File.



The following two “Print To File” dialog boxes will then appear. Press OK for each.



After performing these steps, a file with a .EJF extension will be created in the labels directory. This is the file that PASS software uses to generate the label.

Using Label templates in PASS 6.0

The previous section explains how to use CALTools to create a label template. Once the label template is created, it needs to be referenced in PASS 6.0 program in order to print a label.

The PASS 6.0 Sequence table needs to be modified in order to cause a label to be printed. The Sequence table contains a series of instructions (Sequence items) to be executed by a Dynalab Analyzer when a program is executed.

The PLABEL Sequence item must be added to the Sequence table in order to print a label. Additionally, the Sequence should be designed so that a label is printed only if the harness is good.

Example Sequence: Print a label only if harness is good

Line	Sequence Item	Parameter	Description
1	TEST	MAIN	Performs a complete Netlist scan of the harness.
2	BER	4	Branch to line 4 if errors were logged
3	PLABEL	3X1-B	Prints a label, using label template file 3x1-B
4	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
5	KWAIT		Waits for the operator to push the Start button
6	REPEAT		Goes to line 1 to repeat Sequence execution

Line 1 TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.

Line 2 BER causes the program to branch to line 4 if errors were logged during TEST.

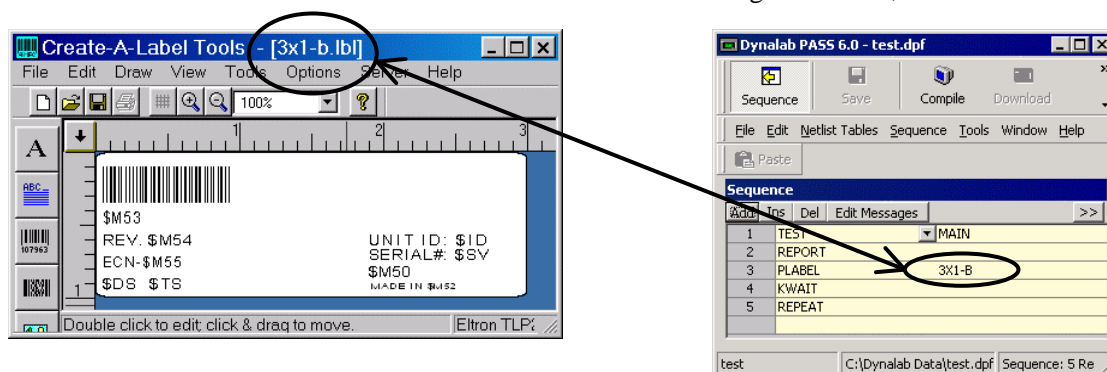
Line 3 PLABEL causes a label to be printed using the label template specified by the Parameter, in this case, 3X1-B. Note that this step is only executed if the harness passes all tests. If errors were logged during TEST, BER in Line 2 would cause PLABEL to be skipped.

Line 4 REPORT displays a summary report

Line 5 KWAIT waits for the operator to press the START button.

Line 6 REPEAT instructs the Analyzer to go to Line 1 and repeat execution of the sequence

Note: The parameter for PLABEL is the name of the label template file that was created using CALTools, as illustrated below:



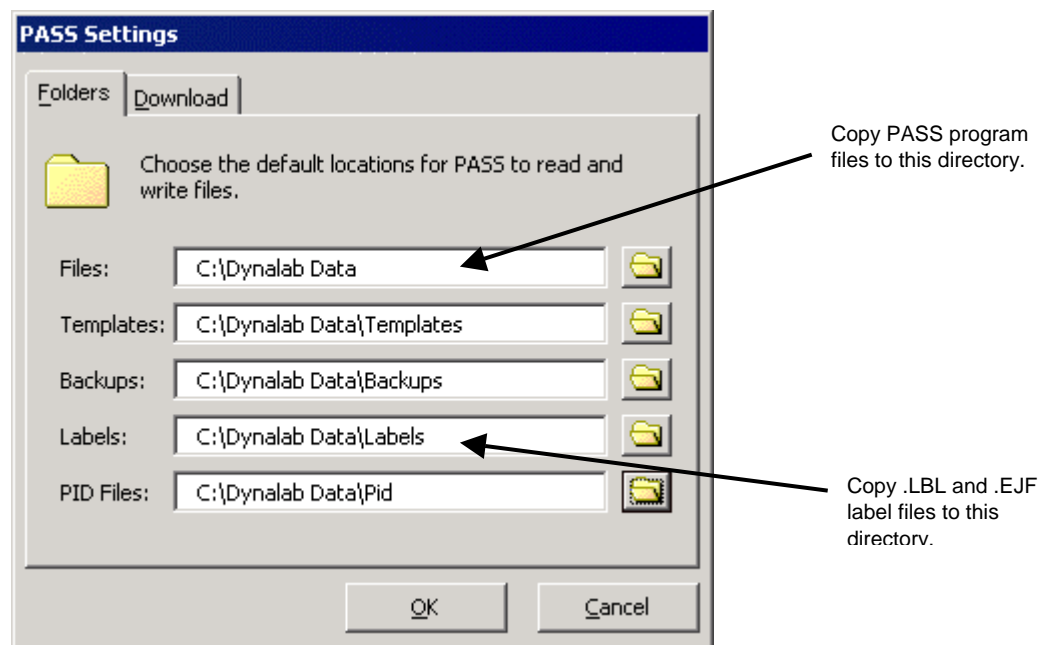
The PLABEL Sequence item may NOT be used more than once in a PASS 6.0 program to print more than one label where each label is based on a distinct label template.

If a second label is to be printed, and if the second label is based on a different label template than the first, the PLABEL2 Sequence item must be used to print the second label.

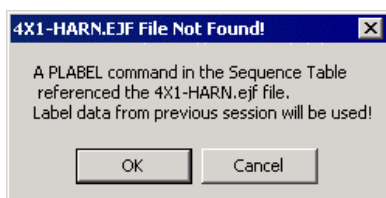
A common label printing application requiring two distinct labels is when a label is to be printed for each good harness, and a different label is to be printed for the box in which the good harnesses are packaged. This example is illustrated in a subsequent section of this document.

Copying label template files and PASS program files from one computer to another

When copying PASS programs from one computer to another, the PASS program must be copied to the target computer's PASS program directory. This directory, as well as other pertinent working directories is specified in PASS 6.0 in the Tools -> PASS Settings screen. If the program prints a label, the corresponding label files with a .LBL extension and .EJF extension should also be copied to the target computer's PASS labels directory.



When the PASS program file and the label .LBL and .EJF files have been copied to the appropriate directories on the target computer, the PASS program must be compiled.



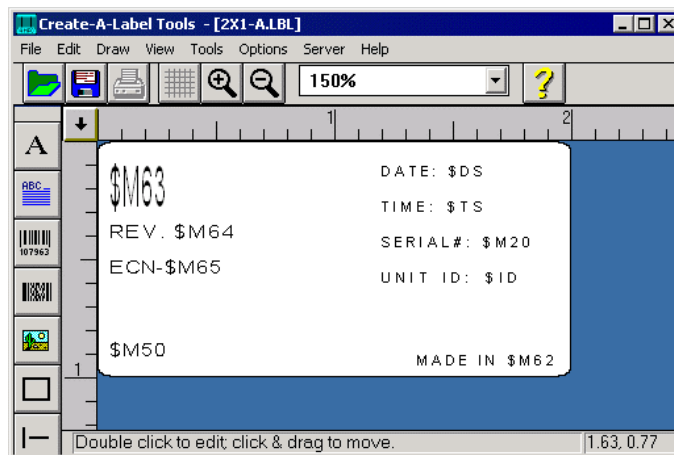
If the .EJF file is not found when compiling, the compiler will indicate this and ask if it is OK to use the label information imbedded in the PASS program file. This may not contain the latest information.

Printing Variable Data

Much of the information that is printed on labels is variable. This section describes variable data and the format symbols used in label templates to represent variable data. Additionally, instructions are provided for printing some frequently used parameters.

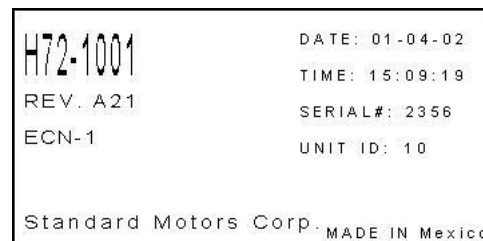
Variable Data and Format Symbols

Consider the pre-defined label template named 2X1-A.lbl. In CALTools, when this file is opened, the following will appear:



The pre-defined label templates contain variable data. The variable data items are represented by format symbols. Each format symbol is identifiable by a leading \$. Format symbols which start with \$M followed by a number refer to message numbers defined in PASS 6.0. So, \$M63 would be message number 63 as defined in the PASS 6.0 Messages table. \$DS is a date stamp, \$TS is a time stamp, \$SV is a counter value, and \$ID is the Unit Identifier. These format symbols are replaced with actual data by the Analyzer when the label is printed.

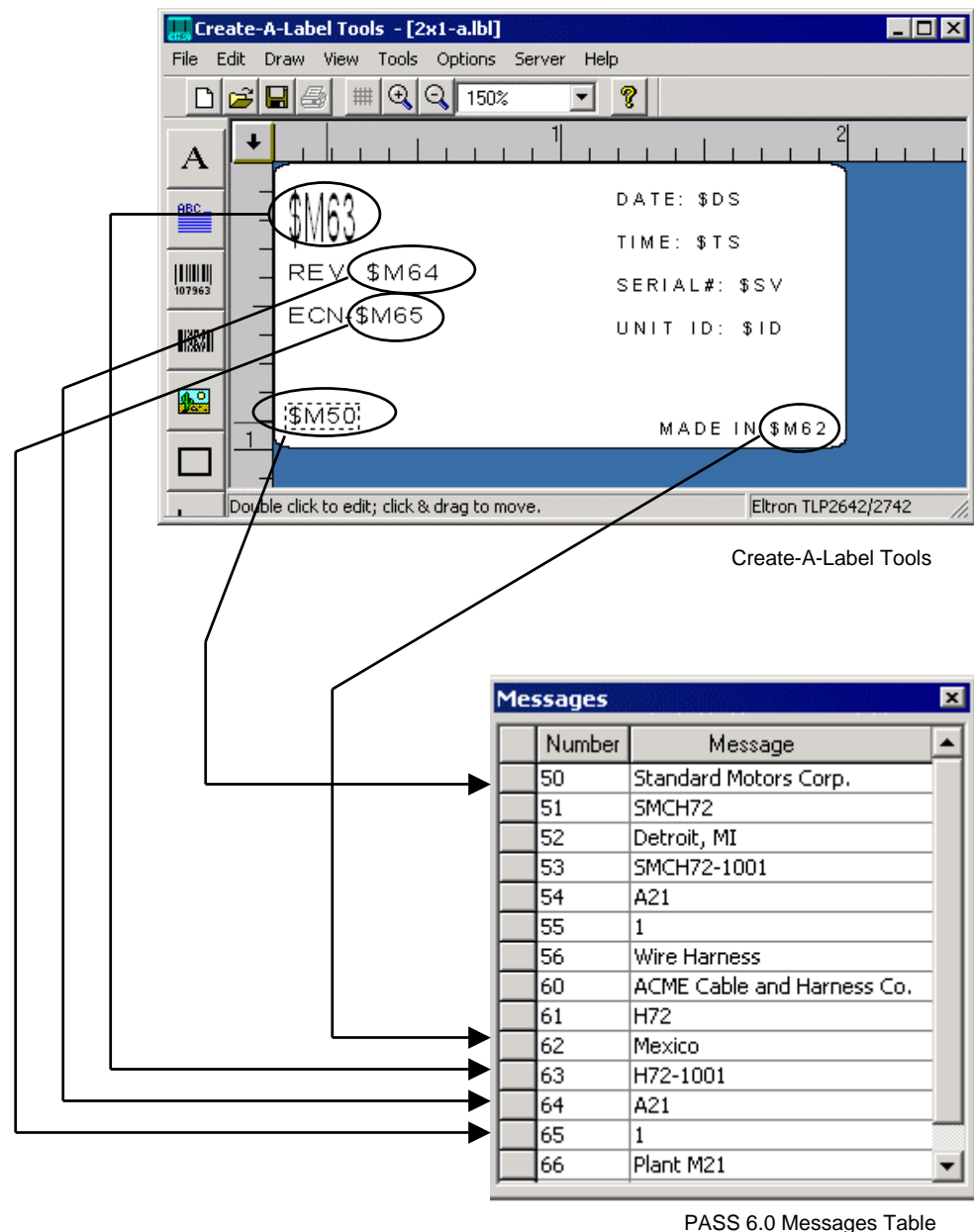
A printed label which was defined using the 2x1-a.lbl label template might look like this.



When the label is printed, the Analyzer substitutes the message text for each of the \$Mnn format symbols. The message text is the corresponding entry in the PASS 6.0 messages table.

For instance, \$M63 was substituted with “H72-001”, \$M64 was substituted with “A21”.

For the other format symbols (\$DS, \$TS, etc), the Analyzer substitutes the appropriate value for each variable.



A complete description of all format symbols that are recognized by PASS 6.0 software and which can be referenced in a label template follows.

This table lists format symbols that can be used in a label template to print time and date information.

Label Template Format Symbols for TIME and DATE

VARIABLE	Label Template Format Symbol	EXAMPLE
Time Stamp	\$ES	Monday, December 25, 2001 10:46:20
Time, hh:mm:ss	\$TS	10:46:20
Hour	\$HR	10
Minute	\$MI	46
Second	\$SC	20
Date, mm/dd/yy	\$DS	12/25/2001
Month Name	\$MO	December
Month Number	\$MM	12
4-digit Year	\$YR	2001
2-digit Year	\$YY	1
Day of the Week	\$DW	Monday
Day	\$DD	25
Julian Date	\$JD	359

This table lists format symbols that can be used in a label template to print miscellaneous information.

Label Template Format Symbols for miscellaneous variables

VARIABLE	Label Template Format Symbol	EXAMPLE
Program Label	\$PL	123HARN
Number of Points	\$PT	122
Unit ID	\$ID	10
Operator ID	\$OP	A231
Holding Register	\$HV	123
Sequence Counter	\$SV	232
Report Counter	\$RV	33
String Variable	\$ST	ABC-123

This table lists format symbols that can be used in a label template to print information about the last error encountered in the test.

Label Template Format Symbols for variable associated with the last error

VARIABLE	Label Template Format Symbol	EXAMPLE
Error Type	\$EE	OPEN
Source Test Point	\$EF	P1-3
Sense Test Point	\$ET	J2-4
Source Wire	\$EC	GRN/YEL (W01)
Sense Wire	\$EX	BLK (W02)

Variable Data and PASS 6.0 Messages Table

In addition to the format symbols listed in the tables above, text messages defined in the PASS 6.0 Messages table may be referenced in a label template. The Label Text is \$Mnn where nn is a number corresponding to the entry in the Messages table. The message number (nn) must be two digits or a single digit preceded by zero.

It is a good idea to standardize the Messages table in PASS 6.0. Although it is not essential, Dynalab recommends standardizing them for ease of maintenance and troubleshooting. These tables show the recommended standard Messages. These are consistent with the pre-defined label templates.

Message Number	Description
50	Customer Name
51	Customer Code / Number
52	Customer Location
53	Customer Part Number
54	Customer Revision
55	Customer Product Level
56	Customer Info 1
57	Customer Info 2
58	Customer Info 3
59	Customer Info 4

Message Number	Description
60	Mfg Name
61	Mfg Code / Number
62	Mfg Location
63	Mfg Part Number
64	Mfg Revision
65	Mfg Product Level
66	Mfg Info 1
67	Mfg Info 2
68	Mfg Info 3
69	Mfg Info 4

Note: Do not use Message ID numbers 80 – 119. They are reserved for internal use by the PASS compiler.

Variable Data and PASS 6.0 Messages: Justification and Padding

The previous sections of this document illustrate how to represent variable data in a label template using label template format symbols. It is useful to note that PASS 6.0 messages also support format symbols. In fact, for every label template format symbol, there is an equivalent messages format symbol.

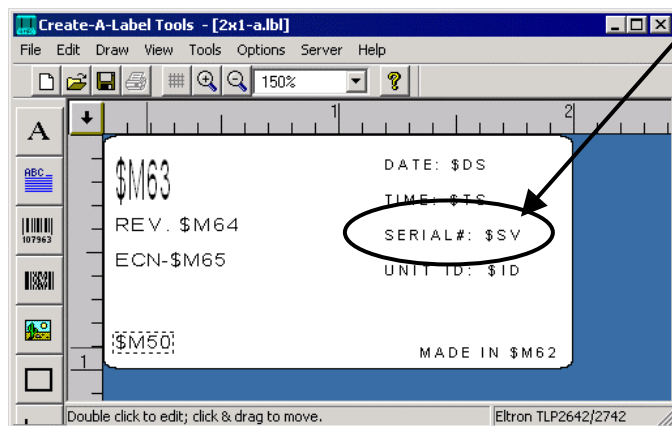
For instance, the label template format symbol for the Sequence Counter is **\$SV**. In PASS 6.0, the messages format symbol for the Sequence Counter is **|q**. So, adding **\$SV** to a label template would be the same as adding a message number such as **\$M15**, if message number 15 in the PASS messages table were defined as follows:

Messages	
Number	Message
15	q

Ordinarily, this would not be very useful – it makes sense to use the direct label template format symbol to represent variable data in a label template. For instance, to print the Sequence Counter, it makes sense to use **\$SV**. It does not make sense to use a label template format symbol representing a PASS 6.0 message number, where the message number contained an imbedded messages format symbol for the Sequence Counter as illustrated above.

However, it is important to understand that the label template format symbols do not allow for any sort of control of justification and padding, but some messages format symbols do. All label template format symbols are left justified.

For instance, if the Sequence Counter is used in the PASS program to represent a serial number, then using the **\$SV** format symbol in a label template would cause the serial number to be printed left justified. In the example below, the text “**SERIAL#:** “ is followed by **\$SV**.



Since label template format symbols are always printed left justified, this part of the label would be printed as follows for several different serial number values:

SERIAL#: 1
 SERIAL#: 456
 SERIAL#: 5378
 SERIAL#: 39874

↑
 LEFT JUSTIFIED

Even though the label template format symbols do not allow for control of justification and padding, some of the PASS 6.0 messages format symbols do. Specifically, the following PASS 6.0 messages format symbols support control of justification and padding:

VARIABLE	PASS 6.0 Messages Format Symbol	CALTools Label Template Format Symbol
Unit ID	u	\$ID
Holding Register	l	\$HV
Sequence Counter	q	\$SV
Report Counter	r	\$RV

To illustrate justification and padding, consider the following examples for printing the serial number, represented by the Sequence Counter. Assume the value of the Sequence Counter is 2378.

Example	Messages Format Symbol	Result
A	q	2378
B	6q	2378
C	06q	002378

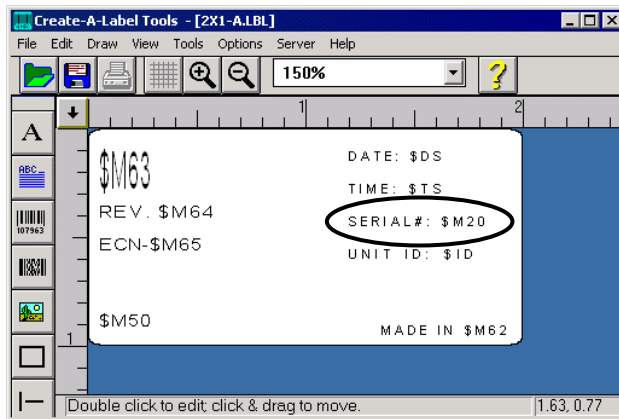
In example A, no control of justification or padding is indicated, so the result is displayed left justified.

In example B, |6q means that the field is 6 characters wide. The result is to display the value right justified with leading spaces.

In example C, |06q means that the field is 6 characters wide, and the leading pad character is zero. The result is to display the value right justified with leading zeros.

The messages format symbols cannot be used directly in a CALTools label template – they can only be used in PASS 6.0 messages. So, to print the serial number using padding and justification, instead of specifying \$SV in the CALTools template, specify a PASS 6.0 message number and define the message in PASS with the appropriate messages format symbol.

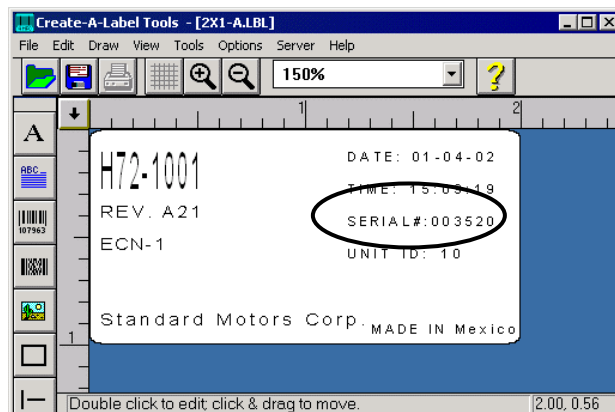
In the example label template shown below, instead of using the \$SV label template format symbol to represent the serial number, \$M20 is used. This means to substitute the contents of message 20 as defined in the PASS 6.0 program's Messages table.



Messages	
Number	Message
20	06q

The entry in the PASS Messages table is shown above. The entry displays the value of the Sequence Counter (|q), right justified in field 6 characters wide, with leading zeros.

The resulting label might look like this. Note the Sequence Counter value is 3520, and is printed in field 6 characters wide. It is right justified in this field, and is padded with leading zeros.



Printing Frequently Used Parameters

This section explains how to generate and print frequently used parameters. Specifically, this section will explain how to generate and print Serial Numbers, Unit ID, and Operator Identification.

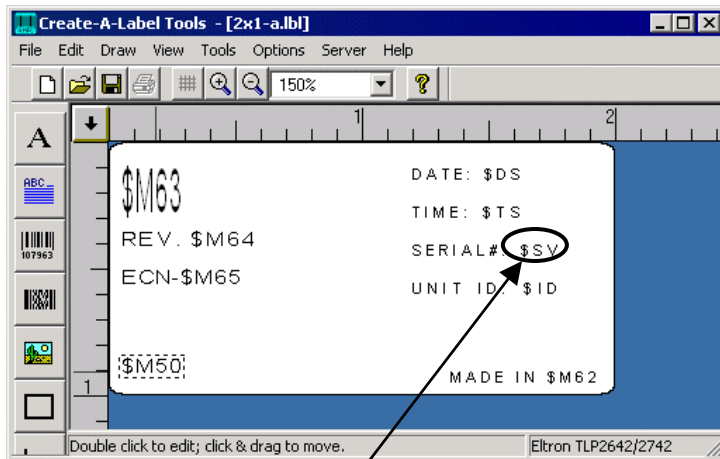
Printing Serial Numbers

A common use for a Dynalab Label Printer is to print a serial number on a label when a harness passes all tests.

The Analyzer has an internal counter that can be used to generate serial numbers. This counter is called the Sequence Counter. The CALTools format symbol for the Sequence Counter is \$SV. In PASS 6.0, the Sequence Counter is symbolized as SCOUNT.

How to generate and print a serial number

To understand how to use the Sequence Counter to generate and print a serial number, consider the pre-defined label template named 2X1-A.lbl. In CALTools, when this file is opened, the following screen will appear:



Note the use of the CALTools format symbol \$SV. This represents the value contained in the Analyzer's Sequence Counter. When the label is to be printed, the Analyzer will substitute the current value of the Sequence Counter for \$SV.

So, the first step in printing a serial number is to include the \$SV format symbol in the appropriate location in the CALTools label template.

The next step in printing a serial number is to include a Sequence item in the PASS 6.0 program that will increment the Sequence Counter every time a harness passes all tests. This is done using the SCOUNT++ Sequence item. SCOUNT++ will increment the Sequence Counter by a value of one.

Here are some important facts about the Sequence Counter:

- Minimum value is 0
- Maximum value is 65535
- The Sequence Counter automatically rolls over to 0 after reaching 65535
- The value of the Sequence Counter is persistent – it does not reset when the Analyzer is turned off or when a new program is selected.

The following is an example PASS 6.0 Sequence, showing the use of SCOUNT++ to generate and print a serial number only if a harness passes

all tests. This example sequence assumes that the label template referenced by PLABEL contains the \$SV format symbol.

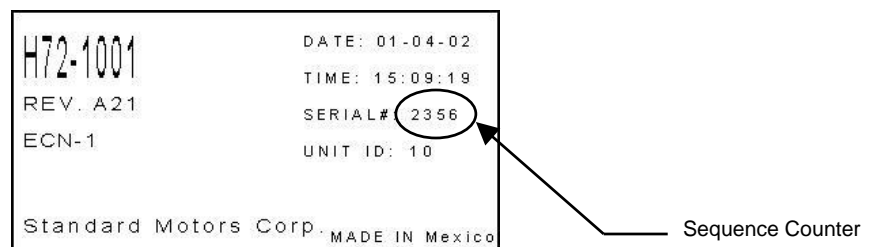
Example Sequence: Print a label containing a serial number only if harness is good

Line	Sequence Item	Parameter	Description
1	TEST	MAIN	Performs a complete Netlist scan of the harness.
2	BER	5	Branch to line 5 if errors were logged
3	SCOUNT++		Increment SCOUNT (Sequence Counter)
4	PLABEL	2X1-A	Prints a label, using label template file 2X1-A
5	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
6	KWAIT		Waits for the operator to push the Start button
7	REPEAT		Goes to line 1 to repeat Sequence execution

- Line 1** TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.
- Line 2** BER causes the program to branch to line 5 if errors were logged during TEST.
- Line 3** SCOUNT++ increments the value of the Sequence Counter by one. Once the Sequence Counter reaches a value of 65535, incrementing by one will cause it to roll over to 0.
- Line 4** PLABEL causes a label to be printed using the label template specified by the Parameter, in this case, 2X1-A.
- Line 5** REPORT displays a summary report
- Line 6** KWAIT waits for the operator to press the START button.
- Line 7** REPEAT instructs the Analyzer to go to Line 1 and repeat execution of the sequence

Note that Line 3 and Line 4 are executed only if the harness passes all tests. If errors are logged during TEST (Line 1), Line 3 and Line 4 are skipped. Therefore, the serial number is incremented and a label is printed only if the harness is good.

Using the pre-defined label template named 2X1-A.lbl, the printed label might look like this:



How to reset the Sequence Counter

As mentioned earlier, the Sequence Counter is persistent. It does not automatically reset when the Analyzer is turned off or when a new program is loaded. The Sequence Counter will automatically reset to 0 after reaching a value of 65535.

There are times when it may be desirable to manually initiate a reset of the Sequence Counter such as shift changes or product changeover.

One way to reset the Sequence Counter is to modify the PASS program such that when the program is initially executed, the operator is prompted to indicate if the Sequence Counter should be reset.

Another way is to manually reset the Sequence Counter through the Analyzer's OPTIONS MENU.

Both methods will be described in this section.

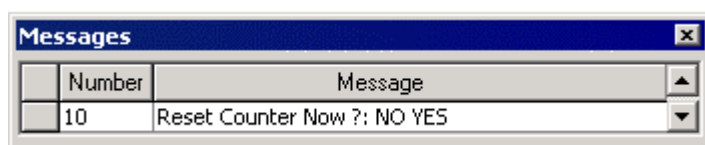
How to modify the PASS program to prompt the operator for reset of Sequence Counter

The following is an example PASS 6.0 Sequence, showing how to prompt the operator to indicate if the Sequence Counter should be reset. When the program is first executed, the Analyzer will display a prompt that says "Reset Counter Now?" followed by two options labeled "NO" and "YES". If the operator selects NO, the program continues using the current value of the Sequence Counter to generate serial numbers. If the operator selects YES, the program will set the Sequence Counter to 0 before proceeding.

Example Sequence: Prompt the operator to reset Sequence Counter

Line	Sequence Item	Parameter	Description
1	MENU	10	Displays a menu as defined in message number 10
2	SWITCH		Go to CASE label matching holding value
3	CASE	2	Label for holding value = 2 (selection is YES)
4	SCOUNT=0		Set SCOUNT equal to zero
5	CASE	1	Label for holding value = 1 (selection is NO)
6	TEST	MAIN	Performs a complete Netlist scan of the harness.
7	BER	10	Branch to line 10 if errors were logged
8	SCOUNT++		Increment SCOUNT (Sequence Counter)
9	PLABEL	2X1-A	Prints a label, using label template file 2X1-A
10	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
11	KWAIT		Waits for the operator to push the Start button
12	GOTO	6	Goes to line 6 to restart execution with TEST

In this example, the PASS Messages table contains an entry for message number 10 as follows:



The screenshot shows a window titled "Messages" with a table containing two columns: "Number" and "Message". The first row shows the number "10" and the message "Reset Counter Now ? : NO YES".

Number	Message
10	Reset Counter Now ? : NO YES

Line 1 The MENU Sequence item displays a menu using the message text associated with the message number specified as the parameter – in this case, 10. Message 10's text is defined as:

Reset Counter Now?: NO YES

This causes the menu and selections to be displayed by the Analyzer as follows:

Reset Counter Now? >NO YES

The operator uses the DOWN and UP buttons to place the cursor next to the desired option. Once the cursor is next to the desired option, the operator presses the START button to make the selection.

If the operator selects the first option (NO), the Analyzer's holding register is set to 1. If the operator selects the second option (YES), the holding register is set to 2.

Line 2 SWITCH causes execution to branch to the CASE label corresponding to the value of the holding register. In this situation, execution branches to Line 5 if the operator selected NO (holding register = 1), or to Line 3 if the operator selected YES (holding register = 2).

Line 3 CASE – since the parameter is 2, this is the CASE label to which execution branches when the holding register = 2. This is where execution resumes when the operator selects YES.

Line 4 SCOUNT=0 causes the value of the Sequence Counter to be equal to zero.

Line 5 CASE – since the parameter is 1, this is the CASE label to which execution branches when the holding register = 1. This is where execution resumes when the operator selects NO.

Line 6 TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.

Line 7 BER causes the program to branch to line 10 if errors were logged during TEST.

Line 8 SCOUNT++ increments the value of the Sequence Counter by one. Once the Sequence Counter reaches a value of 65535, incrementing by one will cause it to roll over to 0.

Line 9 PLABEL causes a label to be printed using the label template specified by the Parameter, in this case, 2X1-A.

Line 10 REPORT displays a summary report

Line 11 KWAIT waits for the operator to press the START button.

Line 12 GOTO 6 instructs the Analyzer to go to Line 6, repeating the testing sequence for the next harness.

How to reset the Sequence Counter by using the Analyzer's OPTIONS MENU

To access the Analyzer's Options Menu from the Main Menu, insert a DynaCard and press the STOP button.

OPTI ONS MENU	
DYNACARD	PROG DATA
MEMORY	>COUNTERS
ERR LOG	AUTOLEARN

In the Options Menu, use the DOWN and/or UP buttons to place the cursor next to the COUNTERS menu item. Press the START button to select the COUNTERS menu item.

COUNTERS	
DI SPLAY	CLR BOTH
CLR REP	SET REP
>CLR SEQ	SET SEQ

In the COUNTERS menu, use the DOWN and UP buttons to place the cursor next to the CLR SEQ menu item. Press the START button to select the CLR SEQ menu item.

CONFIRM	
NO	
>YES	

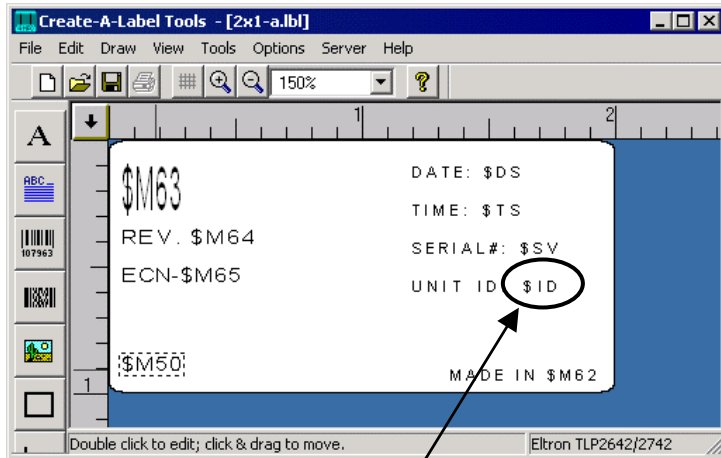
To confirm reset of the Sequence Counter, place the cursor next to the YES menu item. Press the START button to select YES.

The Sequence Counter is now reset to zero. Press the STOP button twice to return to the Main Menu.

Printing the Unit ID

The Analyzer supports a Unit ID function that allows the setting of an identification number from 0 to 255. If several Analyzers are used, each Analyzer can be set to a unique Unit ID. The Unit ID can be printed on a label as a way of identifying which Analyzer tested the harness. The CALTools format symbol for the Unit ID is \$ID.

To understand how to print the Unit ID on a label, consider the pre-defined label template named 2X1-A.lbl. In CALTools, when this file is opened, the following screen will appear:



Note the use of the CALTools format symbol \$ID. This represents the value of the Analyzer's Unit ID. When the label is to be printed, the Analyzer will substitute the value of its Unit ID for \$ID.

So, to print the Unit ID, include the \$ID format symbol in the appropriate location in the CALTools label template.

CALTools Format Symbol
for Unit ID

The Unit ID is set on the Analyzer through the Setup Menus. The Setup Menus are accessed by turning the Analyzer OFF and inserting a DynaCard, then turning the Analyzer back ON while depressing the STOP button.

SETUP MENUS	
>FORMATS	DIAG
RESTRICT	PORTS
CONFIG	

In the SETUP MENUS, use the DOWN and/or UP buttons to place the cursor next to the FORMATS menu item. Press the START button to select the FORMATS menu item.

OUTPUT FORMATS	
ERR TONE	CLOCK
SYMBOLIC	>UNIT ID
LANGUAGE	VOLUME

In the OUTPUT FORMATS menu, use the DOWN and/or UP buttons to place the cursor next to the UNIT ID menu item. Press the START button to select the UNIT ID menu item.

ADJUST UNIT ID:	001
-----------------	-----

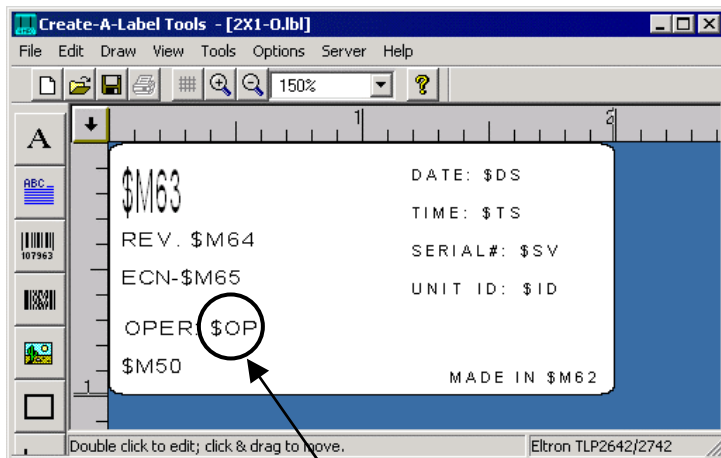
Use the UP and DOWN buttons to adjust the UNIT ID to the desired value between 000 and 255. Press the START Button to move between columns. When the desired value is displayed, press the STOP button to set the Unit ID.

To exit the SETUP mode, turn the Analyzer OFF and then turn the Analyzer back ON.

Printing Operator Identification

The Analyzer supports a capability that allows for the identification of the operator. The Operator ID can be printed on the label. The CALTools format symbol for the Operator ID is \$OP.

To understand how to print the Operator ID on a label, consider the following label template:



Note the use of the CALTools format symbol \$OP. This represents the value of the Operator ID. When the label is to be printed, the Analyzer will substitute the current value of the Operator ID for \$OP.

So, to print the Operator ID, include the \$OP format symbol in the appropriate location in the CALTools label template.

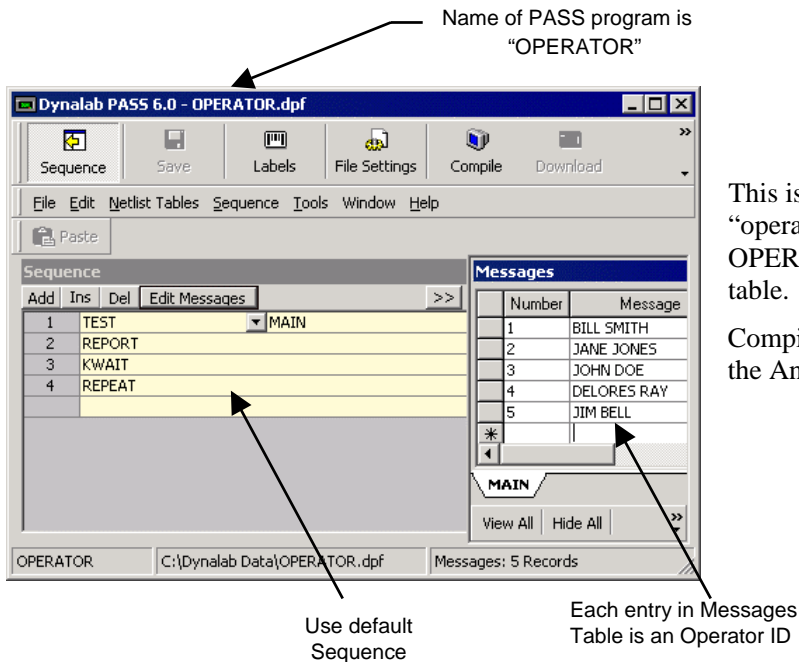
In order to set the Operator ID on the Analyzer, a PASS program needs to be created specifically for this purpose. This specific-purpose PASS program defines the list of Operator ID's. The actual harness test PASS program must then be modified to allow for selection of the Operator ID from the list. Once the Operator ID is selected from the list, it is set in the Analyzer's memory. Then, if the harness test program has a PLABEL Sequence item, and if the label template includes the \$OP format symbol, the Operator ID which is stored in the Analyzer's memory will be printed.

Here is the specific procedure:

- 1 Create a PASS program named "OPERATOR"
 - Enter each Operator ID as an entry in the Messages table. The Operator ID can be a name or a number
 - Use the default Sequence that is created automatically with the program. The Sequence will not be executed, but it must exist in order to successfully compile the program.
- 2 Compile and download this program to the Analyzer
- 3 Modify the harness test PASS program to include the OPERATOR Sequence item at the beginning
- 4 Compile and download the modified harness test program to the Analyzer.

- 5 When the OPERATOR Sequence item is executed in the harness test program, the first Operator ID in the “OPERATOR” Messages table will be displayed.
- 6 Press the UP or DOWN buttons to scroll through the list of predefined Operator Ids
- 7 Press the START button to select the Operator ID

This procedure is illustrated in the following example:



This is an example of a PASS program named “operator”. It has a single Sequence item: OPERATOR. It also has five entries in the Messages table. Each entry is the name of an operator.

Compile the “operator” program and download it to the Analyzer.

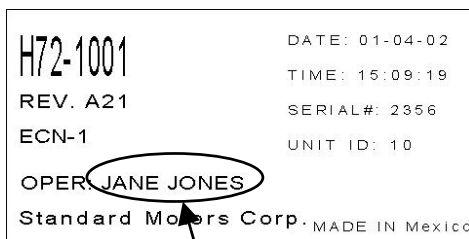
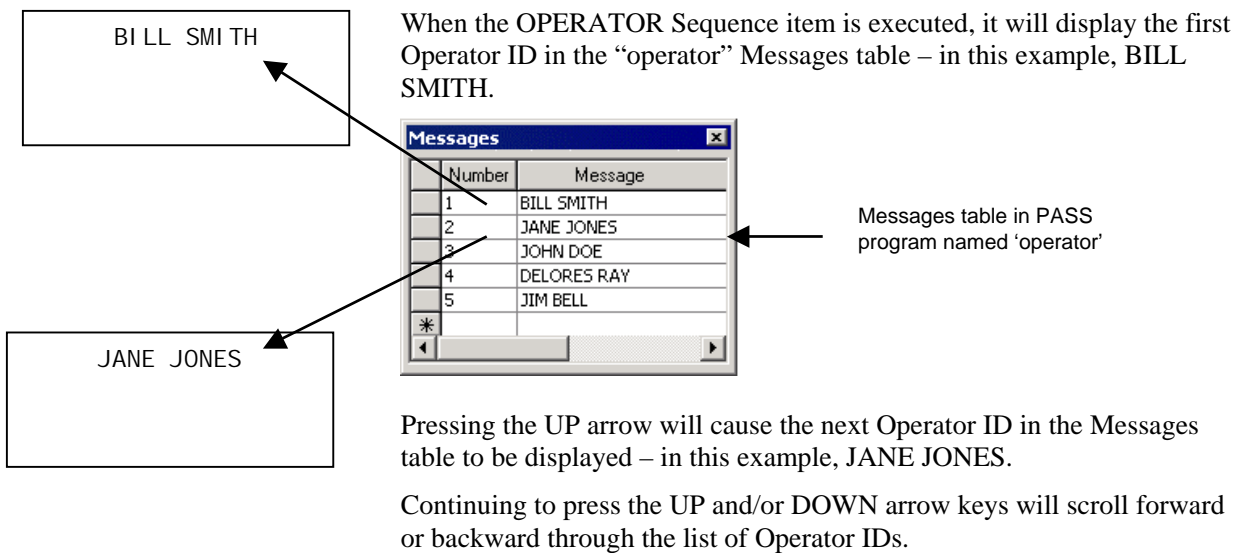
Next, modify the actual harness test program to include the OPERATOR Sequence item at the beginning:

Line	Sequence Item	Parameter	Description
1	OPERATOR		Displays Operator ID list for selection
2	TEST	MAIN	Performs a complete Netlist scan of the harness.
3	BER	6	Branch to line 6 if errors were logged
4	SCOUNT++		Increment SCOUNT (Sequence Counter)
5	PLABEL	2X1-O	Prints a label, using label template file 2X1-O
6	REPORT		Displays “Assembly OK” if harness passes all tests, or displays error information
7	KWAIT		Waits for the operator to push the Start button
8	GOTO	2	Goes to line 21 to repeat Sequence execution at TEST

Line 1 The OPERATOR Sequence item displays the Operator ID list for selection. This list is defined in the Messages table of the program named “OPERATOR”.

Line 2 TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.

- Line 3** BER causes the program to branch to line 6 if errors were logged during TEST.
- Line 4** SCOUNT++ increments the value of the Sequence Counter by one. Once the Sequence Counter reaches a value of 65535, incrementing by one will cause it to roll over to 0.
- Line 5** PLABEL causes a label to be printed using the label template specified by the Parameter, in this case, 2X1-O.
- Line 6** REPORT displays a summary report
- Line 7** KWAIT waits for the operator to press the START button.
- Line 8** GOTO 2 instructs the Analyzer to go to Line 2



Operator ID is substituted for \$OP format symbol when printed.

Once the desired Operator ID is displayed, press the START button to make the selection. The displayed Operator ID is now set in the Analyzer's memory and will be printed by a subsequent PLABEL Sequence item that references a label format containing the \$OP format symbol as illustrated by the example shown at left.

Although this method of defining and selecting an Operator ID may be useful, it does have some drawbacks:

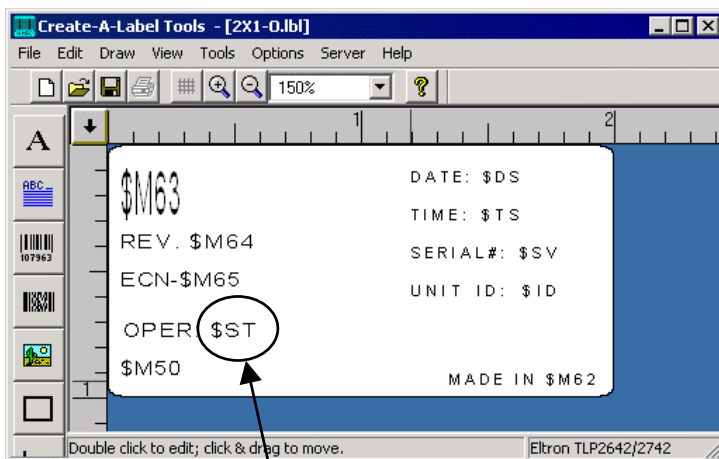
- As changes in personnel take place, the Messages table of the 'OPERATOR' PASS program must be kept up to date.
- There is no guarantee that the operator will select the correct ID.

A better approach may be to identify the operator by scanning a bar coded information on an employee's ID badge. This approach is discussed in the next section.

Printing Scanned Bar Code Information

When equipped with a Dynalab Bar Code Scanner, the Analyzer can read bar coded information. This information can represent almost anything and can be used in a number of ways. This section illustrates how bar coded information read by the Analyzer's Bar Code Scanner can be printed.

In addition to the hardware and software listed in the REQUIREMENTS section of this document, a Dynalab Bar Code Scanner is required. This can be either the Automatic Bar Code Scanner or the Triggered Bar Code Scanner. In either case, the scanner is equipped with a serial cable and a control port adapter. The control port adapter must be connected to the Analyzer's control port. The control port adapter also has a special connector that plugs into a mating connector on the serial cable. Once these connections are made, the Bar Code Scanner is ready for use.



CALTools format
symbol for string of
characters read from
serial port.

Consider an example where the Bar Code Scanner is used to read the operator's identification from a bar code located on the operator's employee ID badge. This information will then be printed on the label.

On the label template shown at left, note the use of the CALTools format symbol \$ST. This represents a string of characters read by the Analyzer on its serial port. Since the Bar Code Scanner is connected to the Analyzer's serial port, data from the Bar Code Scanner is represented in CALTools with format symbol \$ST.

So, the first step in printing scanned bar code information is to include the \$ST format symbol

in the appropriate location in the CALTools label template

The next step in printing scanned bar code information is to include a Sequence item in the PASS 6.0 program that will read a string of characters from the Analyzer's serial port. This is done using the STRING Sequence item.

The following is an example PASS 6.0 Sequence, showing the use of STRING to read and print scanned bar code information - in this case, the operator identification. This example sequence assumes that the label template referenced by PLABEL contains the \$ST format symbol.

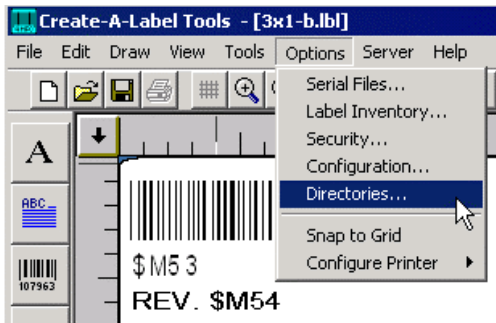
Line	Sequence Item	Parameter	Description
1	MESSAGE	10	"Scan Employee ID Badge Now"
2	STRING		Read string variable from serial port
3	TEST	MAIN	Performs a complete Netlist scan of the harness.
4	BER	7	Branch to line 7 if errors were logged
5	SCOUNT++		Increment SCOUNT (Sequence Counter)
6	PLABEL	2X1-O	Prints a label, using label template file 2X1-O
7	REPORT		Displays "Assembly OK" if harness passes all tests, or displays error information
8	KWAIT		Waits for the operator to push the Start button
9	GOTO	3	Goes to line 21 to repeat Sequence execution at TEST

- 1 A message will be displayed, telling the operator to scan their employee ID badge.
- 2 STRING reads a string of characters from the Analyzer's serial port. In this example, the STRING is reading characters from the Bar Code Scanner. After STRING is invoked, the Analyzer will wait until the bar code is scanned. After the bar code is scanned, the string of characters is stored in the Analyzer string buffer and program execution continues with the next sequence item.
- 3 TEST performs a series of scans on the Netlist specified by the Parameter, in this case, MAIN.
- 4 BER causes the program to branch to line 7 if errors were logged during TEST.
- 5 SCOUNT++ increments the value of the Sequence Counter by one. Once the Sequence Counter reaches a value of 65535, incrementing by one will cause it to roll over to 0.
- 6 PLABEL causes a label to be printed using the label template specified by the Parameter, in this case, 2X1-O. Since the label template contains the \$ST format symbol, the contents of the Analyzer's string buffer will be substituted for \$ST when the label is printed. In this example, the content of the Analyzer's string buffer was populated in step 2 with scanned bar code information representing the employee's identification.
- 7 REPORT displays a summary report
- 8 KWAIT waits for the operator to press the START button.
- 9 GOTO 3 instructs the Analyzer to go to Line 2

Although this example shows how to scan and print the operator's identification, the principles illustrated apply to any scanned bar code information. STRING is always used in the PASS 6.0 Sequence to read a string of characters received from the Bar Code Scanner., and \$ST is the CALTools format symbol which is always used to print the characters that were read by STRING.

Printing Labels Containing Graphics

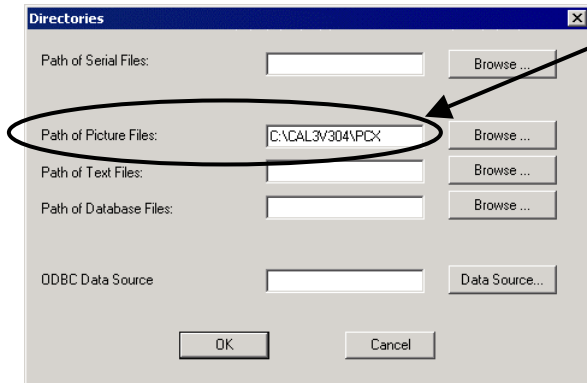
So far, we have discussed how to print labels containing text information. It is also possible to print a label that contains graphics. CALTools supports 45 different graphics formats, including BMP, JPEG, GIF, and PCX. For a complete list, consult the Online Help Manual in CALTools.



Adding a graphics object to a label format

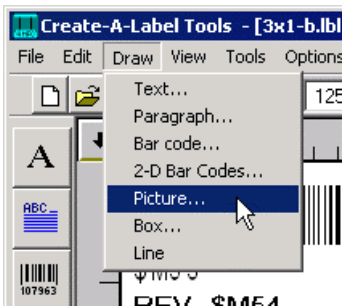
To add a graphical object to a label format, it is first necessary to put a copy of the graphics file in the appropriate directory, so that CALTools can locate it.

In CALTools, select Options->Directories as shown at left.



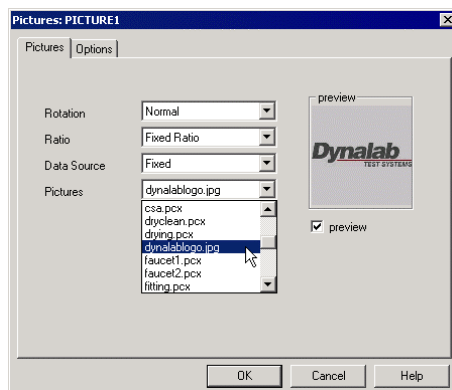
The Directories window will show the location where CALTools expects to find graphics files (Picture Files)

The default location is C:\CAL3V304\PCX. If this is suitable, no changes are necessary. If a different directory is desired, simply enter the directory full path in the "Path of Picture Files" window, or use Browse to locate the desired directory.

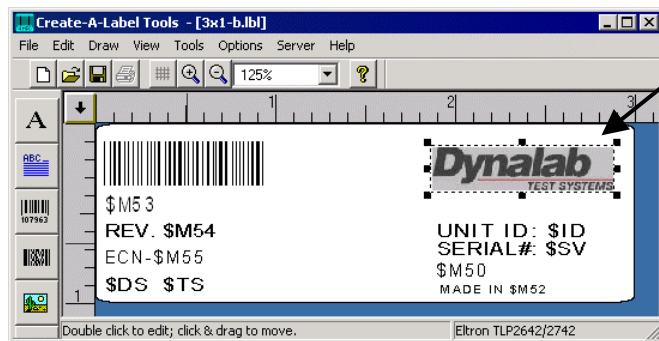


Once the directory for CALTools graphics files is set up as shown above, place a copy of the graphics file to be imbedded in a label format in that directory. The graphics file is now available to CALTools.

Next, in CALTools open the label format file in which the graphics object is to be placed. To insert the graphics object, select Draw->Picture as shown at left.



In the Pictures window, select the desired graphics file from the available files located in CALTools' Picture Files directory. Highlight the desired graphics file, and select OK. This will allow the graphics object to be placed in the label format.

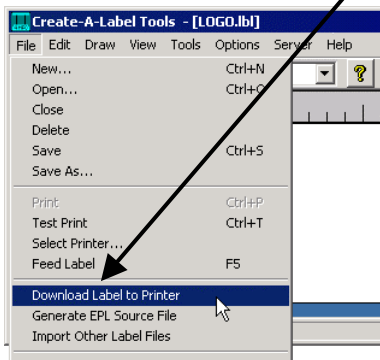


Place the graphics object in the desired location in the label format. Once the label format is set up as desired, save the file and generate the EPL Source File.

Downloading a label containing graphics to the printer

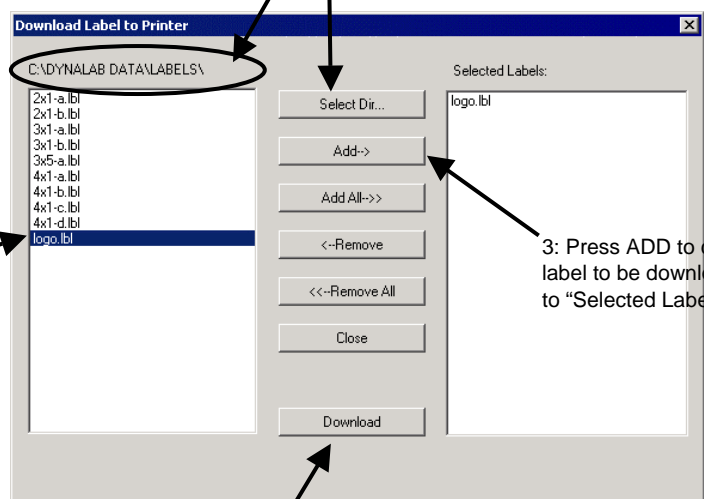
Label format files that contain graphics must be downloaded to the printer's internal memory. So, the final step is to download the label format that contains graphics to the printer as follows:

- 1 Connect one of the host computer's serial ports (COM ports) to the printer with the cable supplied with the printer.
- 2 Make sure that the Eltron printer defined on the host computer is associated with the COM port used in step 1
- 3 Turn power on to the printer.
- 4 In CALTools, select File->Download Label to Printer
- 5 In the Download Label To Printer window, select the directory where the labels are stored, highlight the label to be downloaded, press Add to copy it to the "Selected Labels" section. Then press the Download button. Note that there is no feedback or acknowledgment provided after the "Download" button is pressed.



1: Use Select Dir... to select Directory where labels are stored

2: Highlight label to be downloaded



3: Press ADD to copy label to be downloaded to "Selected Labels" list

4: Press Download to download label format to Printer

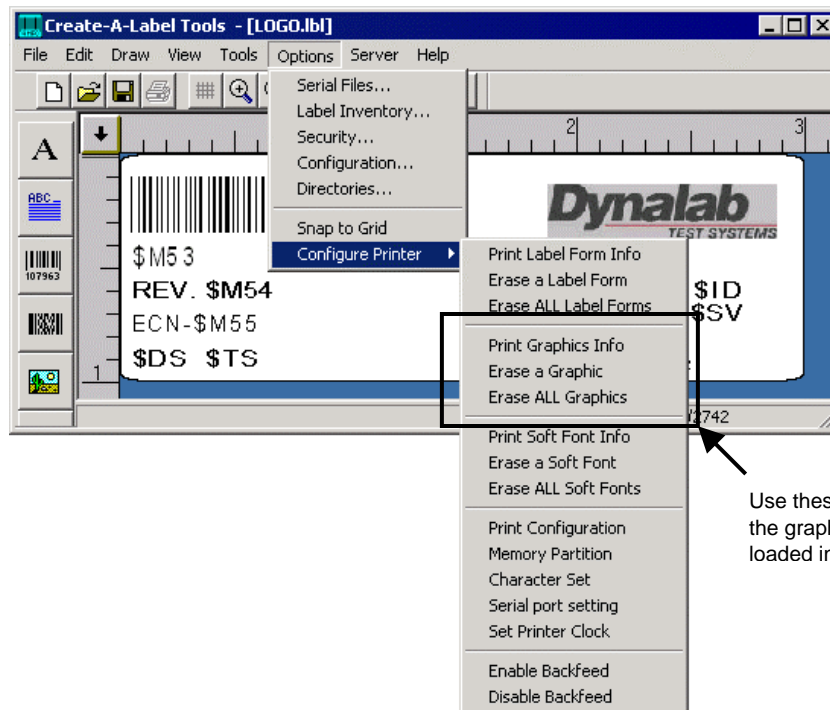
Note: Downloading a label containing graphics usually needs to be done only once. Subsequent changes to the label format design do not ordinarily require that the label be re-downloaded. However, if the graphic is removed from the label template and subsequently re-inserted, the label file must then be downloaded again.

Troubleshooting and managing printer graphics files

The printer has a limited amount of memory available for graphics storage. If after downloading a label format containing graphics, the graphics portion does not print properly, it is possible that the printer's memory allocated for graphics storage is full.

The graphics information stored in the printer's memory may be interrogated and managed by selecting Options->Configure Printer and using the three functions shown below:

- Print Graphics Info
- Erase a Graphic
- Erase ALL Graphics

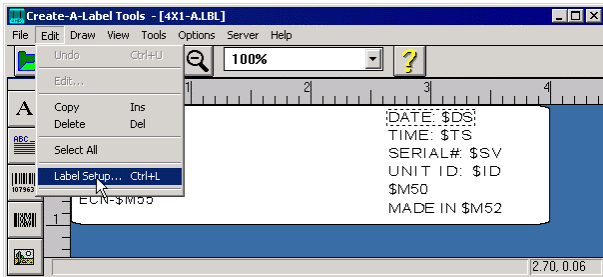


Use these functions to manage the graphics information loaded in the printer's memory

Troubleshooting Common Label Printing Problems

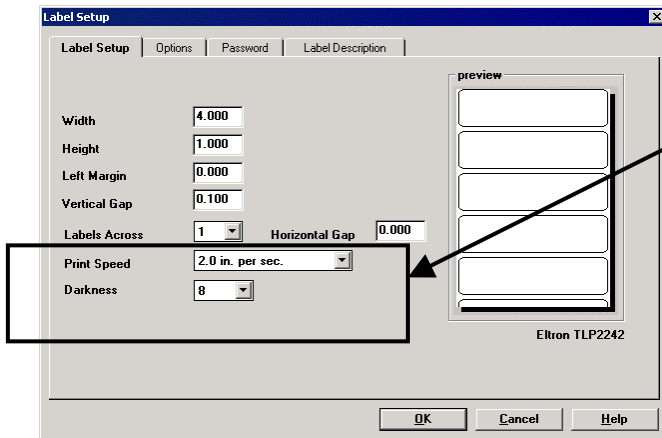
This section describes how to handle common label printing problems.

Problem: Printed label is too light



The intensity of the label ink can be adjusted in CALTools by selecting Edit -> Label Setup as shown at left.

The label Setup screen will be presented.



Two parameters are available on this screen that can affect the intensity of the label ink:

- Print Speed
- Darkness

Ink intensity can be increased by decreasing the Print Speed setting and/or increasing the Darkness setting.

Problem: Objects are not printing

This problem will occur if True Type fonts have been used. In CALTools, make sure that True Type fonts have not been used in the label template. If True Type fonts have been used, change the template to use only fixed fonts.

Problem: Subscript Out of Range Compiler error

When compiling a PASS program that prints a label, the following error is displayed:



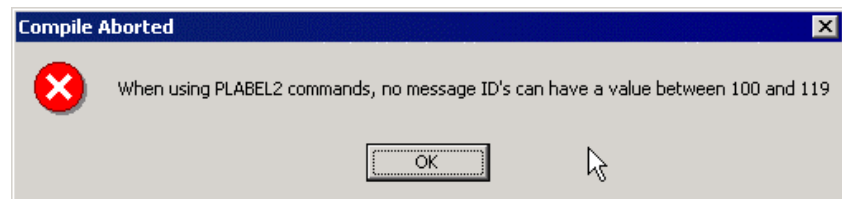
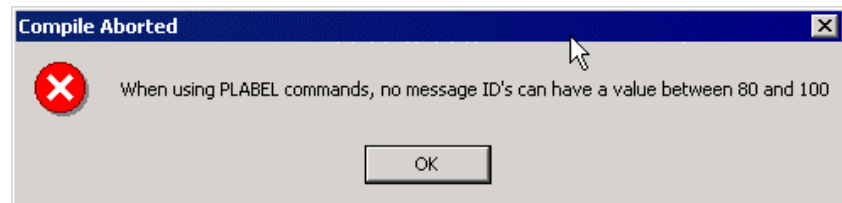
This compiler error will occur if too many objects have been defined in the label template. (An object is a text field, line, bar code, box, etc). The fix for this problem is to re-design the label template in CALTools and use fewer objects.

Problem: Objects do not appear on label exactly as drawn

CALTools does not provide an exact rendering of how objects will actually be printed. It may be necessary to iteratively re-arrange the objects in CALTools and print the label until the objects appear in the desired locations.

Problem: Illegal message ID numbers

When compiling a PASS program that prints a label, one of the following errors is displayed:



The fix to this problem is to avoid using message ID's between 80 and 119. These IDs are reserved for compiler usage.