

# **Schneider PLC Label-It for Concept Readme Version 1.0**

## **Table of Contents**

[About Schneider PLC Label-It for Concept](#)

[System Requirements](#)

[General Usage](#)

[How to Use](#)

[Program Limitations](#)

[How this Program uses the Concept ASC file.](#)

[Resizing Labels](#)

[Debug Troubleshooting](#)

[Contact Information](#)

## About Schneider PLC Label-It for Concept

[Back to Contents](#)

This program will automatically create labels for Schneider Electric I/O modules by reading a Concept converter exported ASC file. Quantum I/O labels are fully supported and the program also allows the user to create 200, 500 and 800 series labels by resizing the standard Quantum label form. The ability to resize these forms also allows the user to fine tune the size of the labels based upon the type of printer they are using.

Figure 1. Example label form

The screenshot shows the 'Schneider PLC Label-It for Concept' software window. The title bar indicates the current page is 17. The interface includes navigation buttons for 'Previous Page', 'Next Page', 'Go to Main Page', and 'Print Page'. Below these are six columns of labels, each representing a different PLC module. Each column has a header 'Outputs' and a list of 16 items, each with a numerical output number and a corresponding module identifier. The modules are: H3D4R1 510, H3D4R1 511, H3D4R1 512, H3D4R1 513, H3D4R1 514, and H3D4R1 515.

Module	Output	Module Identifier
H3D4R1 510	1	0.01
	2	0.02
	3	0.03
	4	0.04
	5	0.05
	6	0.06
	7	0.07
	8	0.08
	9	0.09
	10	0.10
	11	0.11
	12	0.12
	13	0.13
	14	0.14
	15	0.15
	16	0.16
H3D4R1 511	1	XSV0401
	2	XSV0423
	3	FSV0401
	4	XE0426
	5	XSV0402
	6	XSV0403
	7	XSV0405
	8	XSV0406
	9	XSV0409
	10	XSV0410
	11	XSV0411
	12	XSV0413
	13	XSV0414
	14	XSV0422
	15	XSV0424
	16	XSV0425
H3D4R1 512	1	FSV0301
	2	FSV0303
	3	XSV0303
	4	XSV0304
	5	XSV0305
	6	FSV0308
	7	FSV0404
	8	XSV0501
	9	XSV0502
	10	LSV0505
	11	FSV0505
	12	FSV0508
	13	FSV0511
	14	FSV0601
	15	LSV0601
	16	XSV0601
H3D4R1 513	1	FSV0602
	2	FSV0604
	3	HSV0608
	4	HSV0611
	5	TSV0611
	6	FSV0627
	7	FSV0628
	8	LSV0701
	9	XSV0705
	10	FSV0707
	11	LSV0707
	12	FSV0801
	13	LSV0801
	14	XSV0801
	15	LSV0901
	16	LSV0901
H3D4R1 514	1	XSV0901
	2	LSV0903
	3	LSV0908
	4	FSV0909
	5	PSV1004
	6	XSV1103
	7	XSV1301
	8	XSV1602
	9	0.153
	10	XSV2114
	11	LSV2602
	12	XSV4202
	13	XY4104
	14	XSV4301
	15	LSV4302
	16	LSV4302
H3D4R1 515	1	0.161
	2	0.162
	3	0.163
	4	0.164
	5	0.165
	6	0.166
	7	0.167
	8	0.168
	9	0.169
	10	0.170
	11	0.171
	12	0.172
	13	0.173
	14	0.174
	15	0.175
	16	0.176

The user would then print the above form, cut out the labels and place each label in the appropriate sleeve of each I/O module.

## System Requirements

[Back to Contents](#)

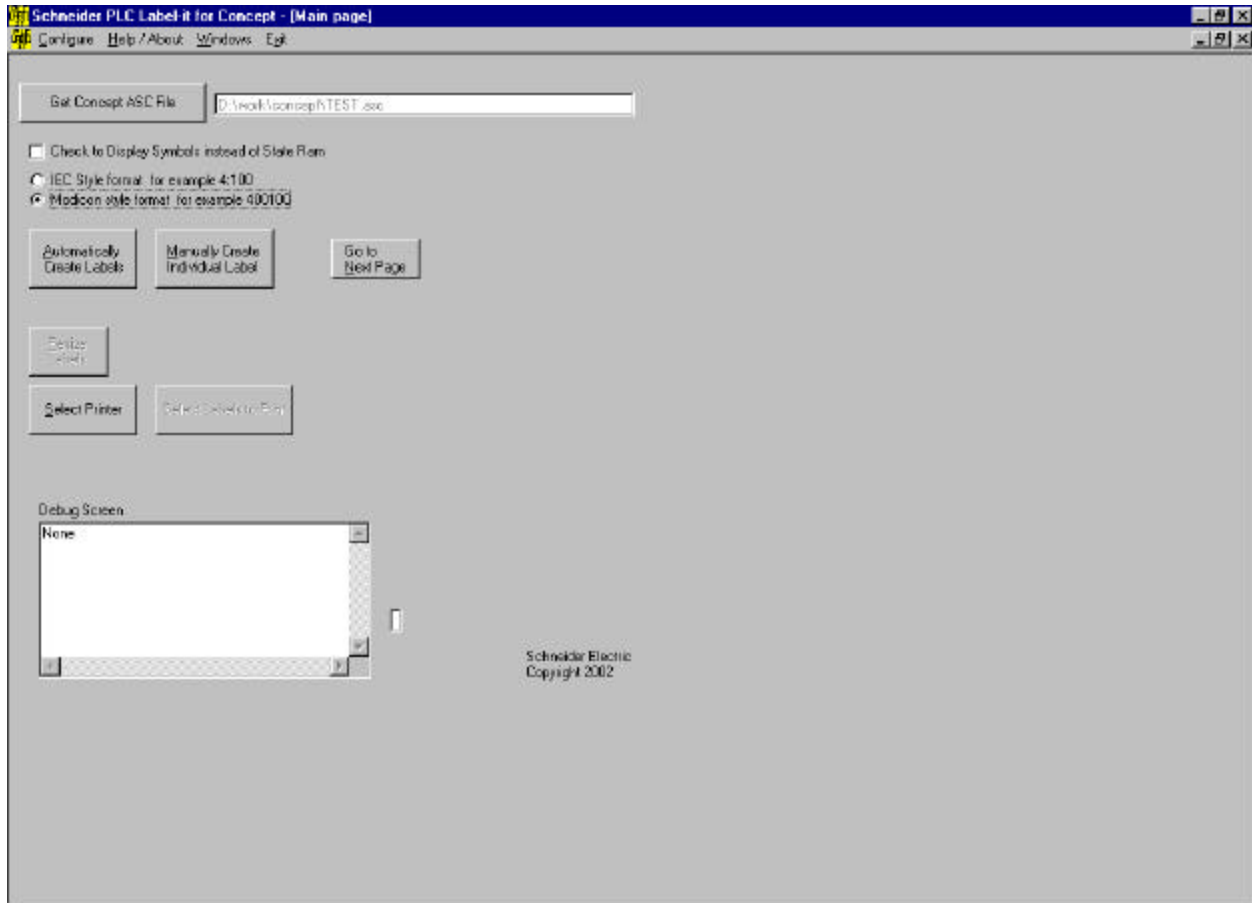
- Microsoft Windows 95, 98 Second Edition, Windows NT 4.0 with Service Pack 5 or 6, Windows 2000, or Windows XP
- RAM Requirements are based upon the amount of labels to create. Approximately 100K to 150K of RAM is required for every 6 labels created. If RAM is limited and the number of labels to create is large then it would be required to edit the Concept ASC file and break the one ASC file into a series of ASC files that would limit the amount of I/O modules per one ASC file.
- 5 MB of available hard disk space for program
- 800 by 600 pixel desktop area or greater
- Printer support
  - This has been tested with LaserJet type printers but will function with bubble jet and other printers capable of printing bitwise graphic images.
  - Concept versions 2.0 and greater ASC files are supported.

## General Usage

[Back to Contents](#)

Figure 2 shows the Label-It for Concept main startup screen. This is the first screen that the user sees when the program is launched.

Figure 2. Startup screen



## How to Use

[Back to Contents](#)

This program uses the Concept Converter ASC file and reads through it to find all of the I/O modules in the I/O map section. It will also read through this ASC file and replace the State ram addressing with the Symbol names (if the user requests).

### Automatically create labels procedure:

1. Use the Concept Converter program ( supplied with Concept ( dbconv.exe) ) export your application program. This can be done with or without DFB's. When the conversion is successfully completed it will generate a text file with the extension of ASC. This is a text file that contains all of the information of your project including the I/O map, logic, symbols etc.
2. Launch the Label-it program and mouse click on the text box. This will popup a dialog box asking for the subdirectory of where the above ASC file is located. Locate it and select Open.
1. If you want to have Symbols on your labels instead of State ram addressing then select and check the "Check to Display Symbols instead of Registers." Checkbox. The user can at this point select whether they want the State Ram to appear in the IEC Format ( 4:100) or in the Modicon format ( 400100 )
2. Click on the Create Labels button. This will start the label creation process.
3. The program will then read through the selected ASC file and create labels for each module found.
4. The user can then Select which pages to print ( from the Main Page) or Print each page individually from the Print button on each page.

### **Manually create labels procedure:**

1. From the main page, when the user clicks on the Manually Create Label button a new blank label page is created.
2. The user, using the mouse, would click on the field that they would like to edit.
3. A popup box appears where the user would enter the desired text.
4. Then the page can be printed.
5. When the user is done they can either delete that page or

### **Program Limitations**

[Back to Contents](#)

The following list identified some of the program limitations:

1. This has been designed to generate labels for Quantum modules however it will generate one label for every module that is in the IO Map, 800 series, Symax, etc. These labels might not be usable to post directly on the IO module but could be posted near the module.
2. This program does not use a database of module parameters. It gathers all of its information from the provided export file. Because of this it does not distinguish between IO module and Specialty modules. For example it will print a label for the high speed counter module 140-EHC-XXX and show that the state ram are inputs and outputs. These labels are indicative of the state ram registers used in the IO map but are not physical inputs or outputs.
3. If a discrete module is mapped to a 3X or 4X register this program must see a module name of 140-Dxxxxxx. Then it knows to map the 3X or 4X register in a bit pattern and will present it in the following nomenclature 4:2 X:0. This would be bit 0 of register 4:2. Thus the 200, 500, 800 series modules will not recognize this difference.
4. This can only fit 32 points on one label. Thus any module with a combined input/output count greater than 32 points will NOT be printed. The program will generate a label and put the module name on the top . The user then can manually type in what they want.
5. The user can not select Modicon format and Symbols. Since the Concept ASC file shows the symbols as referred to the IEC format State RAM.
6. The Total number of characters per I/O parameters that can be changed in the Configure menu item is only used when each label is first created. It establishes the total number of characters that are allowed in each box. If this is changed from the default of 17 to a value that exceeds how many characters will actually print in each I/O box then the user would only see partial text in each particular box.

### **How this program uses the Concept ASC file**

[Back to Contents](#)

The Concept ASC file stores the IO map information using a series of command keywords.

#### BEGIN DropHead

This starts the IO map section and describes the HEAD end.

#### BEGIN IODrop

This starts the individual drop information

#### BEGIN IOModule

This is the individual IO Module section that describes the actual module.

```
outputBytes = 2  
outputReference = 0:1  
inputBytes = 2  
inputReference = 1:1  
moduleName = "140-DAO-840-10"
```

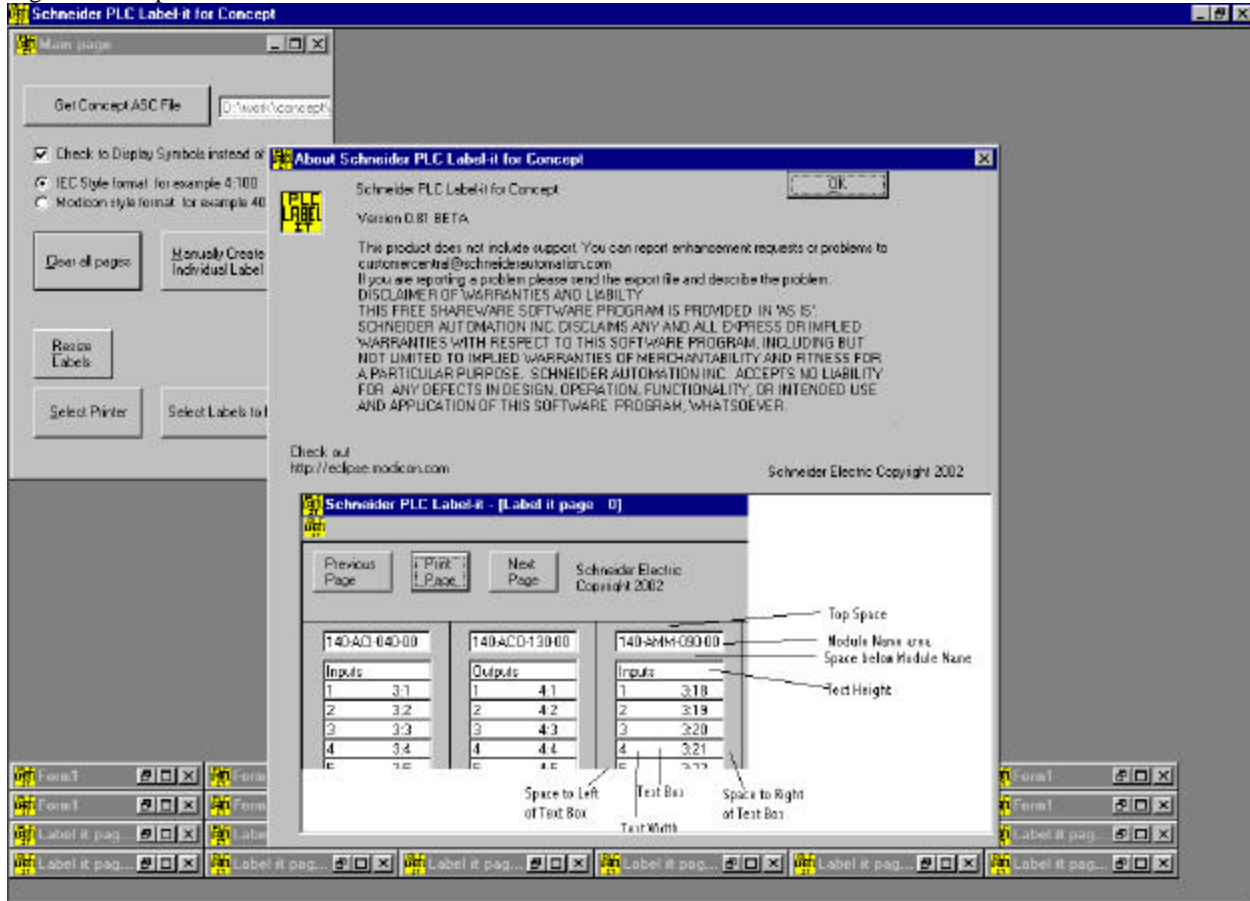
## Resizing labels

[Back to Contents](#)

The Label-It program allows you to change the relative sizing of the labels that have been created. This allows the user the ability to fine tune the resulting printout which will accommodate different printer types.

The user also has the ability to change the spacing of a number of different parameters. If you select the Help/About screen you will see a diagram that explains the terminology of what parameter adjusts which spacing.

Figure 3. Help/About Screen



To adjust any of these parameters change them through the main menu item CONFIGURE .

To resize the textspace in each box you must adjust the value and then create the labels.

## Debug / Troubleshooting

[Back to Contents](#)

At the bottom of the main screen is a debug screen. This text box will allow the user to scroll through to see what the program has accomplished. If any errors have occurred a description will be available in this box.

## Contact Information

[Back to Contents](#)

There is no support available on this product. Any requests for program enhancements or product issues can be emailed to [customercentral@schneiderautomation.com](mailto:customercentral@schneiderautomation.com). If any problems are encountered with this program please email your Concept ASC file, along with a short description of what problems you found to the above mentioned email address.