



Ceaseless Creativity for a Sustainable World

# SATO XML-ENABLED 3.0

## Reference Guide

Version 1.1



© Copyright 2005 SATO Corporation.

All rights reserved. No part of this document may be reproduced or issued to third parties in any form whatsoever without the express permission of SATO Corporation. The material in this document is provided for general information and is subject to change without notice. SATO Corporation assumes no responsibilities for any errors that may appear.

Oracle® is a registered trademark of Oracle Corporation. SAP® is a registered trademark of SAP AG.

All other trademarks are the property of their respective owners.

**OVERVIEW**

<b>1. COMPANY PROFILE</b>	<b>4</b>
<b>2. OVERVIEW</b>	<b>5</b>
<b>3. CHAPTER 1 – INTRODUCTION</b>	<b>6</b>
Introduction	7
Procedure	8
<b>4. CHAPTER 2 – PRINTER FUNCTION – XML DATA STREAM</b>	<b>9</b>
Document Type Definition	10
Sample of XML Data Stream	11
XML/SBPL Mode Selection In Advanced Mode	12
Storage Media Select In Advanced Mode	13
Format Delete in Advanced Mode	14
<b>5. CHAPTER 3 – INTERFACE SOLUTION</b>	<b>16</b>
SATO Printing Solution for SAP, Oracle WMS and Mobile Supply Chain Application and other leading ERP applications	17
How/When to Choose the Best Approach	19
<b>6. CHAPTER 4 – NICELABEL PRO</b>	<b>20</b>
Overview	21
Using NiceLabel Pro to Create Label Format	22
Export Label Format to XML-Enabled Printer	26
<b>7. GLOSSARY</b>	<b>29</b>



## COMPANY PROFILE

SATO developed the world's first printer capable of printing barcodes and OCR characters. SATO went on to become the leader in the Data Collection System and Labeling Solutions Industry providing its customers (from different industries such as the medical, packaging and entertainment industries) in many countries with revolutionary products that helped improve customer productivity, efficiency & cost savings.

SATO has continued to strive to produce even more innovative products over the years: direct thermal and thermal transfer printers, OEM print engines, printer accessories, label design software, labels and thermal transfer ribbons. Today, SATO still continues with its belief and tradition of designing and manufacturing high quality industrial printers that are not only reliable and durable but also delivers superb performance with each print job.

## OVERVIEW

The SATO XML-enabled Printer Reference Guide contains the following sections:

- 1. *Introduction*** - The SATO XML-enabled printer is introduced in this chapter to provide a highlight of some of its features.
- 2. *Printer Function XML Data Stream*** – In-depth information exploring the benefits of printing barcode labels with SATO’s XML-enabled printers.
- 3. *Interface Solution*** - Implementation of SATO XML-enabled printing for SAP, Oracle® WMS and Mobile Supply Chain Application (MSCA) and other leading ERP applications.
- 4. *NICELABEL PRO*** - SATO XML-enabled label formats can be created using NiceLabel Pro.
- 5. *Glossary*** - Provides a list of terms and definitions for your reference



## CHAPTER 1

# INTRODUCTION

The SATO XML-enabled printer is introduced in this chapter to provide a highlight of some of its features.

## INTRODUCTION

### Why XML?

Information in XML format provides greater flexibility as it can directly integrate with various systems. XML format offers flexibility as it is not confined to limited preset tags, which is contrasting to other mark-up languages with preset tags such as HTML. In other words, XML allows each vendor in a specified industry to create its own set of tags to cater to that industry without much sacrifice of compatibility issues. Comparing stored information is the traditional approach, where for example, a third party software, typically the middleware, is required to store the data in an acceptable format into the database.

Now, large scale database systems, such as Oracle Warehouse Management System (WMS), can store information in XML format.

### Advantage with SATO XML - Parser printer

1. Provide direct printing solutions with any database system without the need for any additional print server hardware and middleware application. The outcome is an overall printing system infrastructure that improves the efficiency of time and reduces unnecessary costs that would have otherwise been used to develop the middleware application.
2. Print labels from the Oracle Warehouse Management System (WMS) infrastructure or any ERP database application which is able to generate label information in XML format.

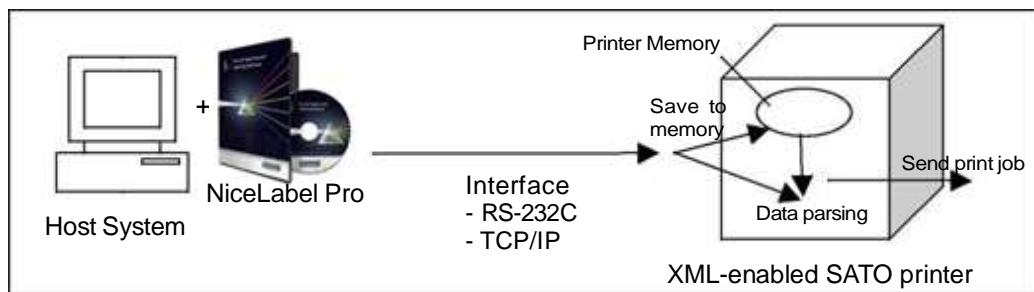
### Printer Memory

The XML-enabled SATO format is stored in a memory card that is residing in the printer. A standard SBPL feature is to download the label formats. The SBPL in the label format for the SATO XML-enabled printer is modified to include the necessary information that allows a link to be established between the XML variable data and the variable field on the label.

## PROCEDURE

1. **The XML-enabled label format is allocated into the printer's memory by the means of a label designing software.** The host system is used to transfer the format data to the printer for storage. The host computer is also used to send XML data stream to the printer to enable label dispense.
2. **The host sends a print job in XML format to the printer over a network via an interface such as TCP/IP or by other means of connection.** The data sent from the host system to the printer is analyzed by the printer's firmware to see if it is SBPL or XML data stream. (Please note that the user is able to set the printer to XML or SBPL mode by going to the printer to parse only SBPL data stream whereas setting the printer to XML mode will allow the printer to parse only XML data stream.
3. **The print job parses the label format number, label quantity and variable field name.** When the XML data stream follows the Document Type Definition (DTD) semantics, it contains a <labels> or <label> element with a \_FORMAT attribute. The XML-enabled printer parses the XML print job data stream for the format attribute, label quantity, and variable field data.
4. **The printer will search for the label format number from its memory. The variable data field name is then inserted accordingly, where the format is recalled and the print job begins (Merging the Format and XML variable data).** When the SATO XML-enabled printer has the requested format (stored in the memory card), it recalls the format by interpreting the SBPL in the format. This will define the layout of the format (i.e. static text), and defines which fields are variables. The printer then applies the variable field data from the XML data stream to the label format.

When end-of-label is detected in the XML data stream, the label is printed.







## CHAPTER 2

# PRINTER FUNCTION – XML DATA STREAM

In-depth information exploring the benefits of printing barcode or RFID smart labels with SATO's XML-enabled printers.

## DOCUMENT TYPE DEFINITION

XML DTD is outlined in this section, defining the implementation of an XML parser. SATO XML-enabled printer will translate the XML according to this DTD. The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared inline in your XML document, or as an external reference.

The following briefly illustrates the purpose of XML.

- XML stands for **E Xtensible Markup Language**
- XML is a **markup language** much like HTML.
- XML was designed to **describe data**
- XML tags are not predefined in XML. You must **define your own tags**.
- XML uses a DTD (**Document Type Definition**) to describe the data.
- XML with a DTD is designed to be **self-describing**.

The printer name, label format, and duplicate quantities are given in the <labels> tag initially to be used as a default value for each individual, <label> within the XML. Those default values can be overridden by each individual <label> tag for a specific printer name, label format, and/or duplicate quantities for that particular label print request. Note that unless values are given within the <label> tag, the attributes (such as printer name, label format, and quantity) will apply through the rest of the XML message.

```

<!ELEMENT labels (label)*>
<!ATTLIST labels _FORMAT CDATA #IMPLIED> <!ATTLIST labels
_JOBNAME CDATA #IMPLIED> <!ATTLIST labels _QUANTITY CDATA
#IMPLIED> <!ATTLIST labels _PRINTERNAME CDATA #IMPLIED>***

<!ELEMENT label (variable)*>
<!ATTLIST label _FORMAT CDATA #IMPLIED> <!ATTLIST label
_JOBNAME CDATA #IMPLIED> <!ATTLIST label _QUANTITY CDATA
#IMPLIED> <!ATTLIST label _PRINTERNAME CDATA #IMPLIED>

<!ELEMENT variable (#PCDATA)> <!ATTLIST variable name CDATA
#IMPLIED>

```

## SAMPLE OF XML DATA STREAM

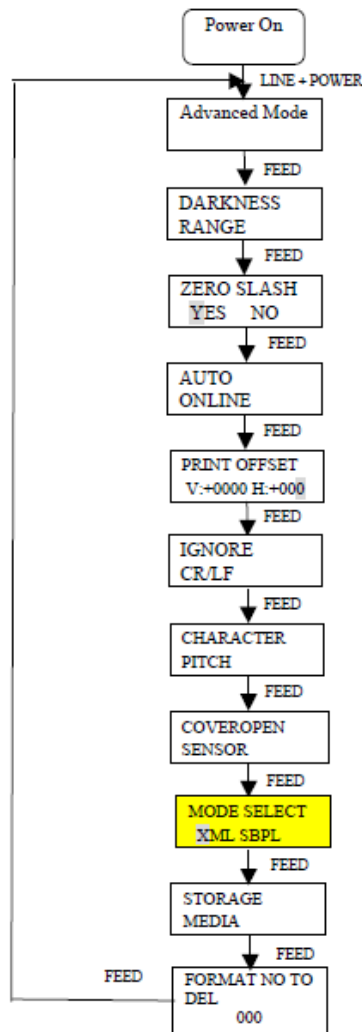
The content of the XML message lies between <labels> and </labels>, which corresponds to a group of one or more label print requests. The content between <label> and </label> has the information to print one individual label. Each <variable> and </variable> pair gives the value of one variable on the label. The printer name, label format, and duplicate quantities are given in the <labels> tag initially to be used as a default value for each individual, <label> within the XML. Those default values can be overridden by each individual <label> tag for a specific printer name, label format, and/or duplicate quantities for that particular label print request. Note that unless values are given within the <label> tag, the attributes (such as printer name, label format, and quantity) will apply through the rest of the XML message.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd"> <labels
_FORMAT="1" _QUANTITY="1"
_PRINTERNAME="Printer1" _JOBNAME="LBL101">
<label>
<variable name= "ITEM">AS5488</variable>
<variable name= "ITEMDESC">Desktop</variable>
<variable name= "QTY">2</variable>
<variable name= "UOM">EA</variable>
</label>
<label _FORMAT="LotLabel" _PRINTERNAME="Printer2" >
<variable name= "ITEM">AS12345</variable>
<variable name= "ITEMDESC">Mouse</variable>
<variable name= "LOT">10200</variable>
<variable name= "QTY">10</variable>
<variable name= "UOM">Bx</variable>
</label>
</labels>
```

As we can see from the above example, the XML data stream contains the <labels> element that defines the format, quantity, printer name, and job name. SATO XML-enabled printer shall use the \_FORMAT element, which only takes numerical digits, to recall the stored XML-enabled format from memory, and also the \_QUANTITY element to determine the number of labels to be printed. Notice that the LOT is one of several defined variable data fields, and the value, 10200, is the variable data to be printed on the label. The printer shall extract these variable data and match them with the pre-defined variables in the Format that is recalled by the printer.

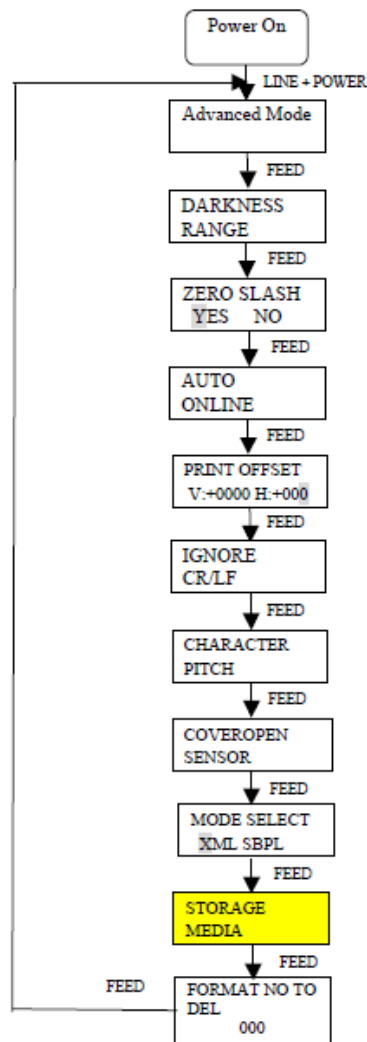
## XML/SBPL MODE SELECTION IN ADVANCED MODE

The figure below is showing the LCD displays for the Advanced Mode of the XML enabled printer. The user is able to select the desired operating mode of the printer by going to the MODE SELECT Option in Advanced Mode. Selecting XML mode will allow the printer to parse XML data stream whereas selecting SBPL mode will allow the printer to parse SBPL data stream. Note that the printer is not able to parse XML data stream when in SBPL mode. Likewise, the printer is not able to parse any SBPL data stream when in XML mode except for the Format Store <ESC>YS and Format Recall <ESC>YR commands. Note that the default mode of the printer after Factory Clear is XML mode.



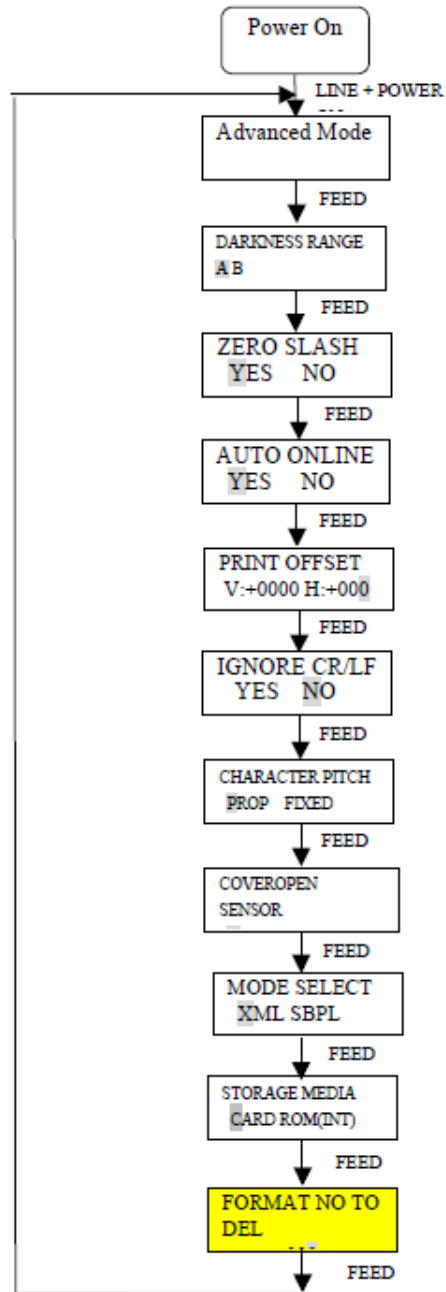
## STORAGE MEDIA SELECT IN ADVANCED MODE

The figure below is showing the LCD displays for the Advanced Mode of the XML enabled printer. The user is able to select the desired storage media by going to the STORAGE MEDIA Option in Advanced Mode. This option will tell the printer from which storage media to retrieve the stored Format when processing XML data streams. Note that the default storage media is the printer's internal Flash ROM.



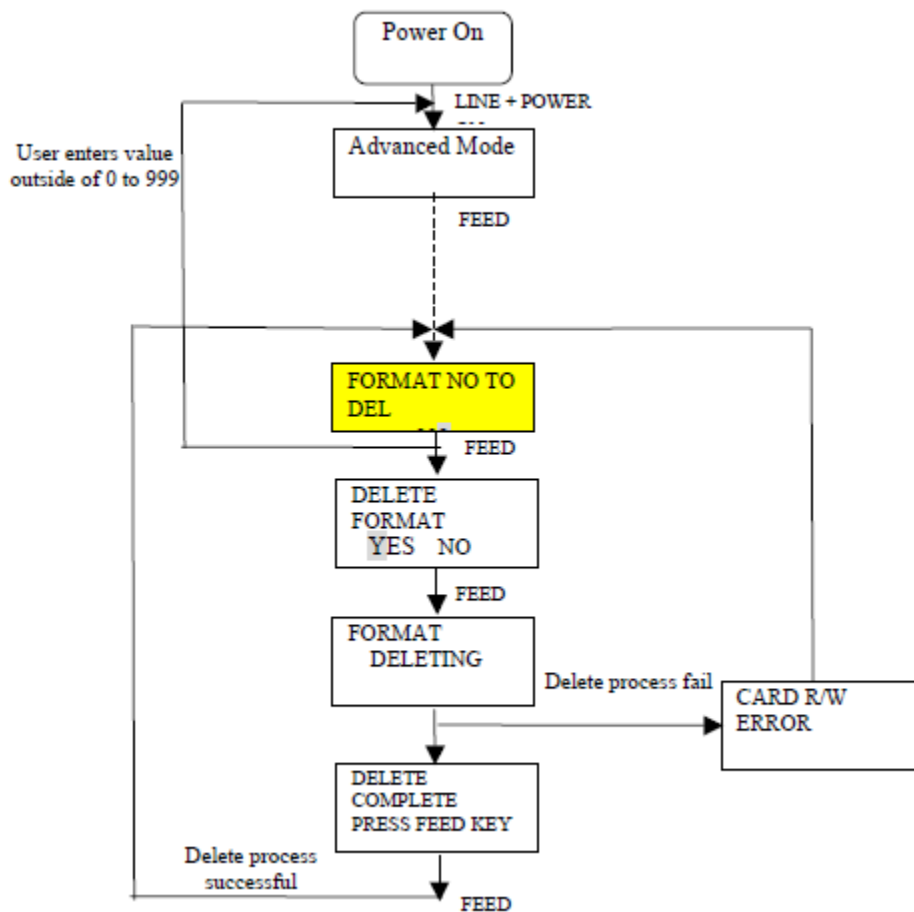
## FORMAT DELETE OPTION IN ADVANCED MODE

This “FORMAT NO TO DEL” option was implemented in the Advanced Mode to allow the user to delete a stored Format in the SRAM card or Flash ROM.



## SEQUENCE OF STEPS TO DELETE A FORMAT

The following figure shows the sequence of steps the user will have to take when deleting a format. After the user has input the Format number in the “FORMAT NO TO DEL” LCD message, the “DELETE FORMAT” LCD message will pop up to prompt the user to confirm whether to delete the format or not. If the user selects “YES”, the delete process will start and the “FORMAT DELETING” LCD message will appear. The printer will delete the format from either the SRAM card or ROM (INT). This is dependent on where the “Storage Media” option in Advanced Mode is pointing to. If the delete process is successful, the “DELETE COMPLETE” LCD message will appear along with three short beeps to inform the user that the delete process is completed. Once the user presses the FEED key, the process will revert back to the “FORMAT NO TO DEL” LCD message. If an error occurs in the middle of Delete process, a “CARD R/W ERROR” LCD message will appear along with a long beep to warn the user that the operation has failed. The process will again revert back to the “FORMAT NO TO DEL” LCD message. Note that if the user enters a format number outside of the range 1 to 999 e.g. 0, the process will go to the “Advanced Mode” LCD message. The user can also use the <ESC>\* command to delete a format.



## CHAPTER 3

# INTERFACE SOLUTION

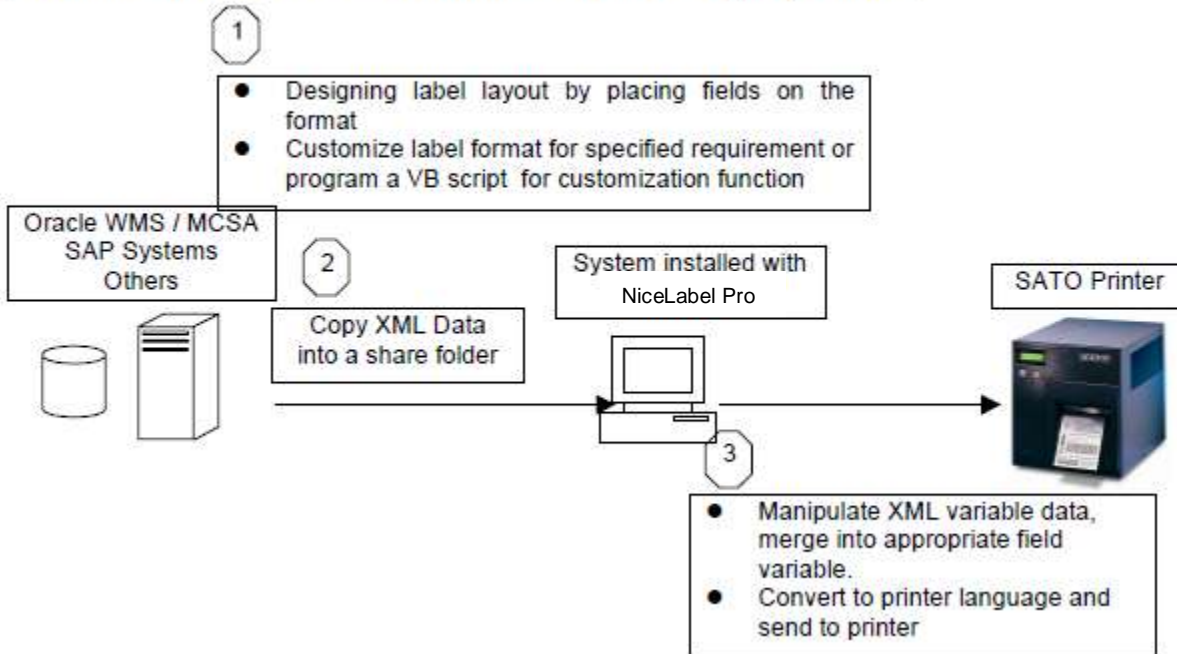
Implementation of SATO XML-enabled printing for SAP, Oracle® WMS and Mobile Supply Chain Application (MSCA) and other leading ERP applications.



## SATO PRINTING SOLUTION FOR ERP APPLICATION

SATO offers two printing solutions interface to ERP application.

### Solution 1: Middleware Solution with SATO Label Designing Software



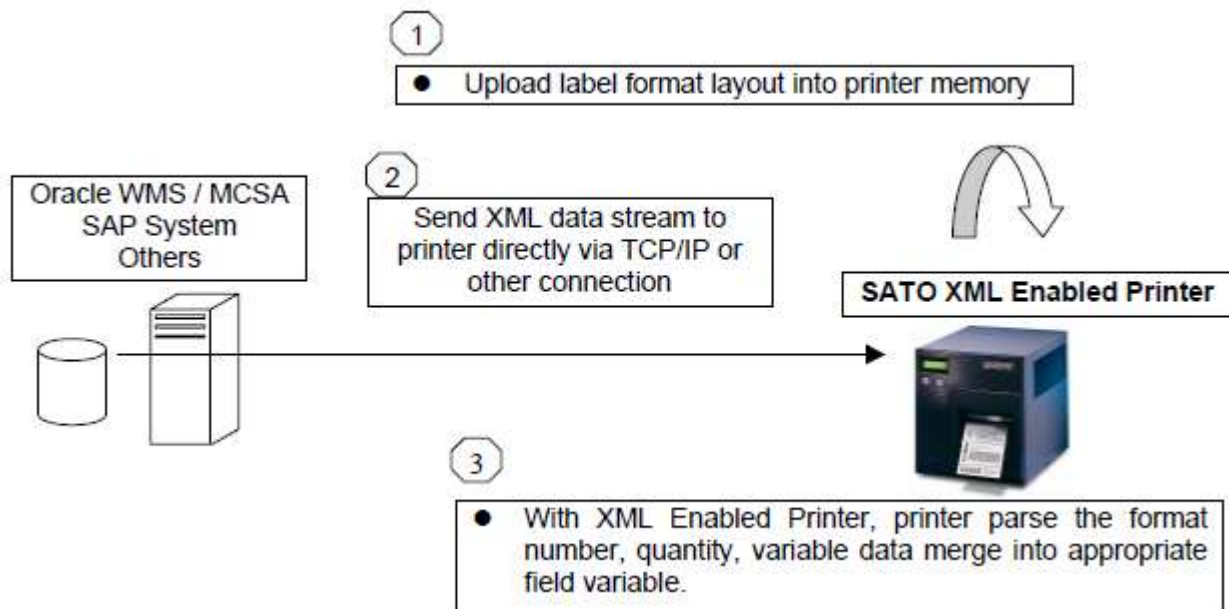
In this approach, Automation Easy acts as a middleware solution to receive a XML data stream from ERP applications. Automation Easy is executed at the workstation. Each time a label printing is triggered in Oracle system or others, a XML file containing the label data is copied to a share directory of workstation. Automation Easy will look at the share directory to check if any changes have been modified in the file. If any changes are detected, it will extract the variable data from the XML file and merge it into the pre-defined label format. The printer driver will then convert it to the printer's language and transmit the data to the printer. NiceLabel Pro can also be used to re-create a specified template format based on customized requests from different customers with differing needs. Hence, customized programming can be done using this "acting" middle layer.

Automation Easy is a software package, which will manipulate the label at the appearance of trigger file from Oracle WMS / MSCA or others. This activity generates a trigger event for Automation Easy to start with action execution.

**Solution 2 : Direct Printing approach with XML Parser**

ERP Applications uses extensible Mark -up Language (XML) as the interface to exchange data with other applications. In order to connect SATO Printer direct to such system such as Oracle WMS/MSCA and others without any additional print server or middleware, SATO print offers you a build-in XML parser firmware inside a printer. With XML parsing function, SATO printer understands the native XML data stream generated by Oracle WMS / MSCA or any leading ERP system

In this approach, the first step is to design the label format in NiceLabel Pro with matching variable names. Next is to register a pre-defined format into printer memory. Up to 999 formats can be registered, but this very much depends on the size of memory. Next, the label information in XML native format is then transfers to SATO printer via TCP/IP, wireless or other connection. With XML parsing function, XML parser will analyses the XML data stream which are the label format number, label quantity, and the variable data on the corresponds variable field. Then printer recalls the format and output the barcode label.



## HOW / WHEN TO CHOOSE THE BEST APPROACH

### **Solution 1: Middleware Solution with NiceLabel Pro Software**

#### Required Items in this approach:

- Customize Java application that execute at Oracle WMS / MSCA or other system to give a triggering signal to the workstation
- NiceLabel Pro software to perform a format designing and execute printing.

#### **Advantage**

- Easy to customize the format layout.
- Filtering support for modifying the XML stream from Oracle WMS / MSCA and other leading ERP applications.
- Provide error notification to user if printing error occurs.

### **Solution 2 : Direct Printing approach with XML Parser**

#### Required Items in this approach:

- SATO Printer with XML Parsing function enabled.

#### **Advantage**

- Fewer layers, which simplifies the whole printing solution and minimizing costs.
- No additional middleware and print server is required, reducing costs.

## CHAPTER 4

# NICELABEL PRO

SATO XML-enabled label formats can be created using NiceLabel Pro.

## OVERVIEW

A purchased copy of NiceLabel Pro can be used to create label formats with variables. Each designed label format can be exported to the SATO XML CL408e printer by classifying format number.

### **Important:**

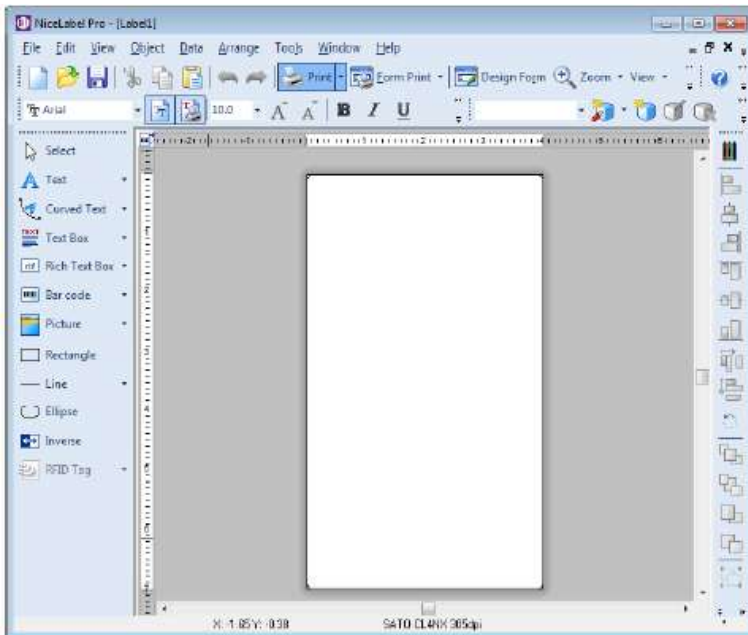
Before designing the label layout, you need to note the followings:

- Please use only SATO internal fonts and barcodes.
- Please design with field name that are exactly same as the XML file and they are case sensitive.

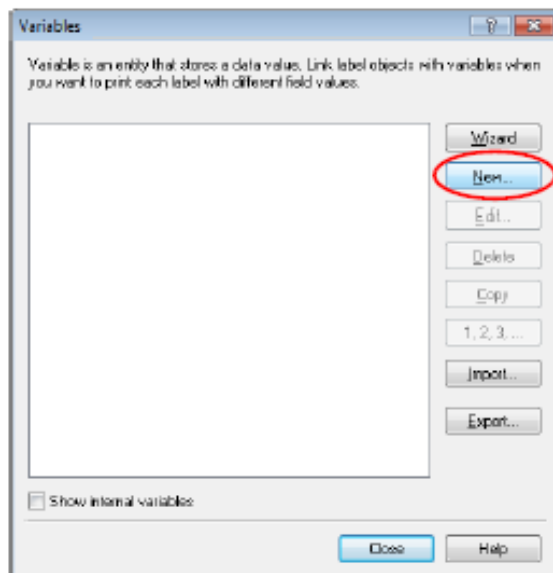
## Using NiceLabel Pro To Create Label Format

### Create Label

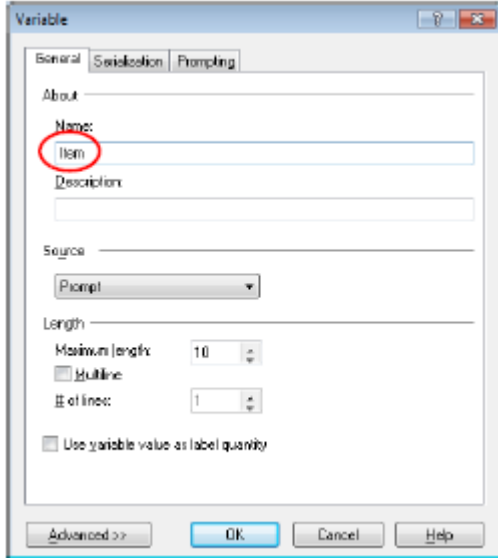
1. Open NiceLabel Pro 6.3 and above
2. From the menu bar, select <File > New Label
3. To add a XML variable data field, select Variables under Data menu in the menu bar.



4. Click the New button on the Variables window.

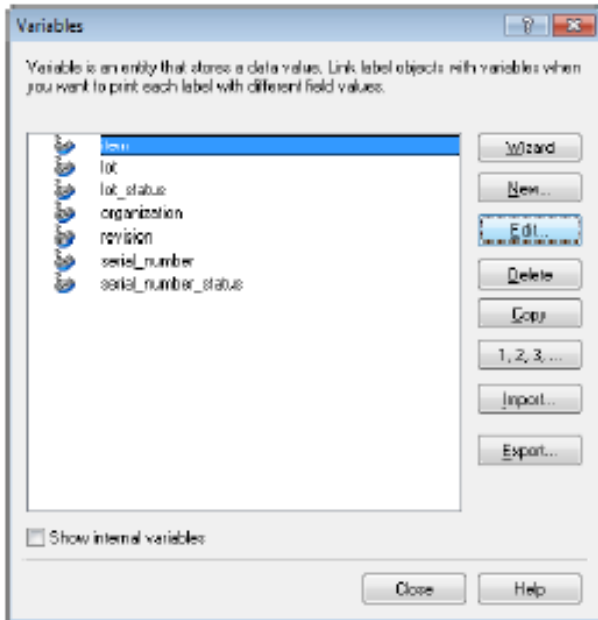


5. On the Variable window, enter the name of the variable according to the desired XML variable field in the XML file (See “Fig. 1: Sample of the XML Data Stream for the serial label format.” On page 24). Then click the OK button.

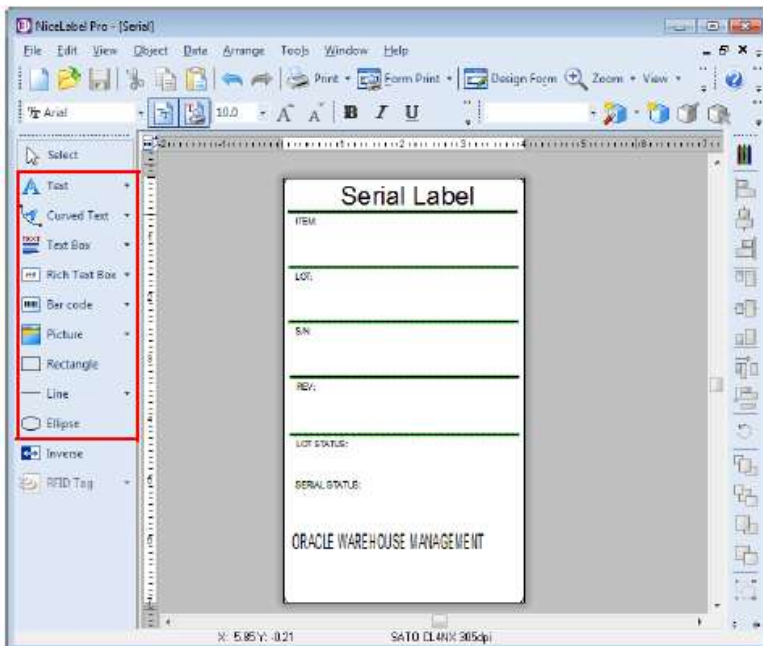


6. For creating additional XML variable, repeat step 4 and 5.

7. To rename variables, return to the **Data > Variables** window, select the variable and click the **Edit** button.

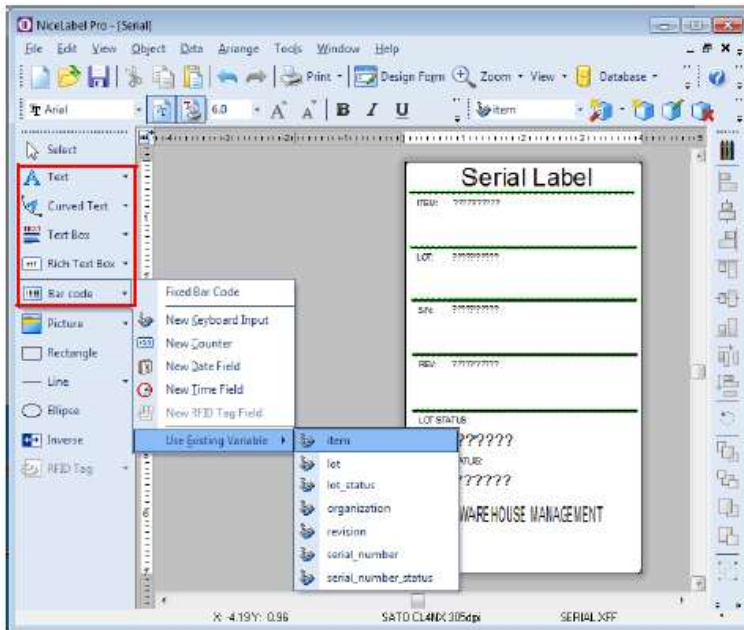


8. To design the rest of the label, first, create the static objects (text, graphics, barcode), which do not change with every label. Refer to the NiceLabel Pro user guide for more details.

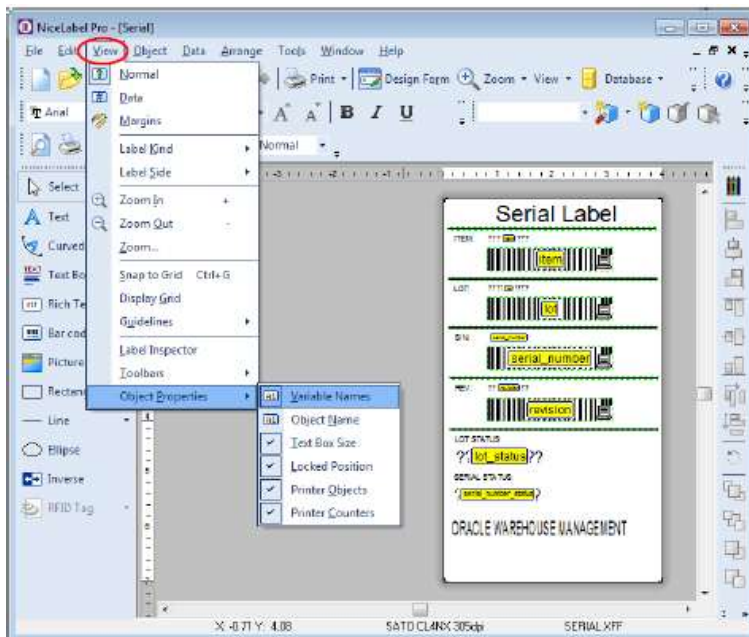




9. To insert XML variable data fields, select **Use Existing Variables** from the various data type drop down list and then select the variable.



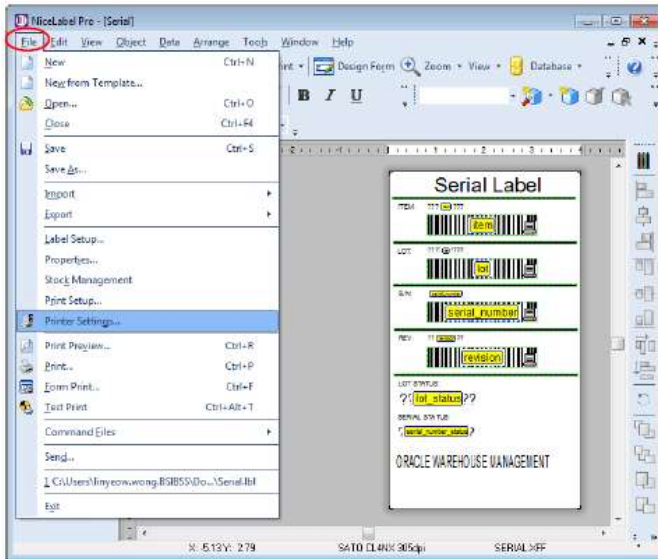
10. You can select **View > Object Properties > Variable Names** to view the variable names on the screen.



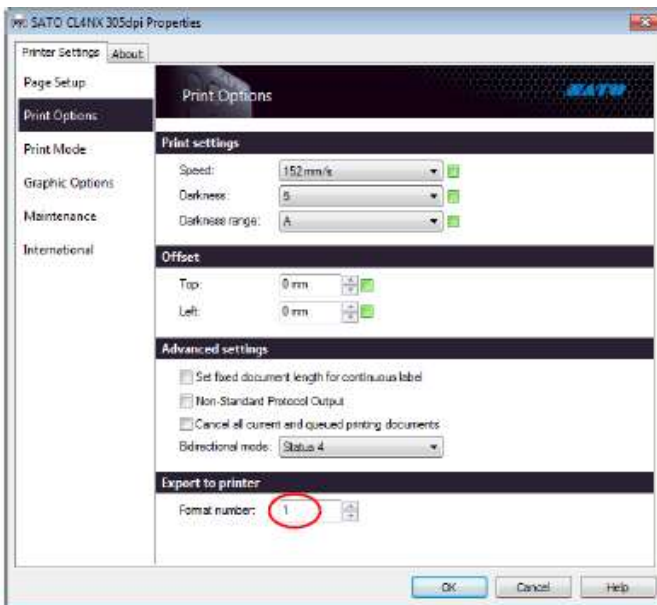
## Export Label Formats To SATO XML Enabled Printer

### To set the format number before exporting to XML printer

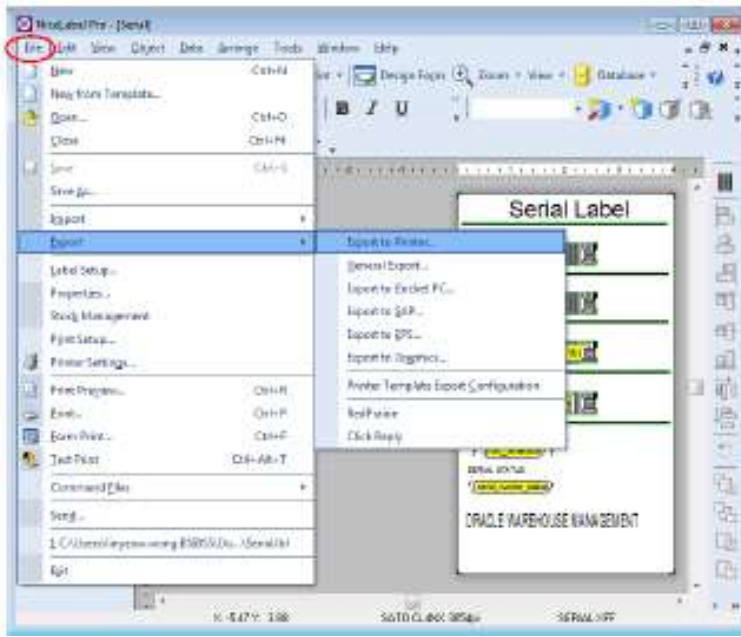
1. After the label format design is completed, select Printer Settings under File menu in the menu bar, or click the printer settings icon to set the format number to be assigned for this label design.



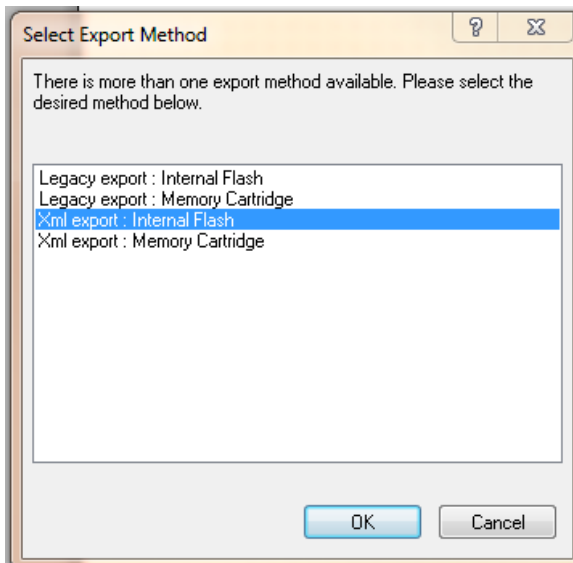
2. On the SATO printer **Properties** window, select the **Print Options** from the **Printer Settings** tab. Set the **Format number** and click **OK** button.



3. Save the label format. The label format is ready to export to the XML-enabled printer.
4. Set the SATO XML-enabled printer to online mode. Select File > Export > Export to Printer from the menu bar.



5. On the Select Export Method window, select XML export: Internal Flash and then click OK button.



```

serial.xml - Notepad
File Edit Format View Help
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels_FORMAT="10"_QUANTITY="1" _PRINTERNAME="Printer 1" _JOBNAME="LBL101">
  <label>
    <variable name = "organization">611</variable>
    <variable name = "item">NUTS</variable>
    <variable name = "lot">1234</variable>
    <variable name = "serial_number">12345</variable>
    <variable name = "revision">V2</variable>
    <variable name = "lot_status">1234</variable>
    <variable name = "serial_number_status">Active</variable>
  </label>
</labels>
  
```

Fig. 1: Sample of the XML Data stream for the serial label format.

```

ESC@ESC@CC2ESC@YS,001ESC@/X,SerialESC@ESC@H0115ESC@V00014ESC@P02ESC@RDA00,48,51,Serial
LabelESC@ESC@H0016ESC@V00506ESC@FW04H0455ESC@ESC@H0016ESC@V00392ESC@FW05H0455ESC@ESC@H0024ESC@V00604ESC@P02ESC@RDA00,16,1
7,SERIAL STATUS:ESC@ESC@H0027ESC@V00523ESC@P02ESC@RDA00,16,17,LOT
STATUS:ESC@ESC@H0014ESC@V00282ESC@FW05H0455ESC@ESC@H0027ESC@V00409ESC@P02ESC@RDA00,16,17,REV:ESC@ESC@H0013ESC@V00176ESC@F
W04H0455ESC@ESC@H0025ESC@V00299ESC@P02ESC@RDA00,16,17,S/N:ESC@ESC@H0025ESC@V00193ESC@P02ESC@RDA00,16,17,LOT:ESC@ESC@H001
1ESC@V00066ESC@FW04H0455ESC@ESC@H0025ESC@V00083ESC@P02ESC@RDA00,16,17,ITEM:ESC@ESC@H0021ESC@V00703ESC@P02ESC@RDA00,21,42,0
RACLE WAREHOUSE
MANAGEMENTESC@ESC@H0099ESC@V00080ESC@GR001ESC@ESC@H0099ESC@V00189ESC@GR002ESC@ESC@H0099ESC@V00295ESC@GR003ESC@ESC@H0099
ESC@V00407ESC@GR004ESC@ESC@H0028ESC@V00545ESC@GR005ESC@ESC@H0024ESC@V00629ESC@GR006ESC@/X,13,10,ItemESC@ESC@H0099ESC@V001
11ESC@BG02048ESC@/X,14,10,lotESC@ESC@H0099ESC@V00219ESC@BG02048ESC@/X,18,10,serial_number:ESC@ESC@H0099ESC@V00327ESC@BG020
48ESC@/X,17,10,revisionESC@ESC@H0099ESC@V00439ESC@BG02048ESC@ZESC@
  
```

Fig. 2: Sample of label format (in SBPL codes) exported to SATO XML-enabled printer.

## GLOSSARY

Provides a list of terms and definitions for your reference

**Data stream** A continuous flow of information or data transmitted on a byte-to-byte basis.

**DDE (Dynamic Data Exchange)** A protocol that enables communication between Windows applications. An application must support DDE in order to use it. With DDE Commands you can control one application (for example NiceLabel Pro) with another (for example Microsoft Access).

**Document Type Definition (DTD)** States what tags and attributes are used to describe content in an XML document, where each tag is allowed, and which tags can appear within other tags.

**Firmware** Computer programming instructions that is stored in a read-only memory unit rather than being implemented through software.

**NiceLabel Automation Pro** NiceLabel Automation Pro is separately purchased software, which acts as middleware solution to receive a XML data stream from ERP applications. Automation Pro will manipulate the label at the appearance of trigger file.

**JOB file** This is a plain ASCII text file containing NiceLabel commands, which are to be executed. JOB files are usually used to print a batch of labels.

**Middle Layer (Middleware)** Software agents acting as an intermediary between different application components. It is used most often to support complex, distributed applications.

**NiceLabel Pro** A barcode design and printing software that lets you create and print labels and fully supports WYSIWYG (What You See Is What You Get). Using NiceLabel Pro, you can easily design any label to include bar codes, text, lines, boxes and graphics.

**Parse** To break down into smaller component parts with an explanation of the form, function and syntactical relationship of each part.

**RS-232C** An interface used for serial communications such as between the printer and the computer for the purpose of light-duty applications. Also known as 'serial cable'

**SBPL** SATO's native command language that allows users to produce high quality labels using its printer control language and label definitions.

**TCP/IP** A protocol for communication between computers, used as a standard for transmitting data over networks and as the basis for standard Internet protocols

**Trigger event** Event that must occur before NiceLabel Automation Pro begins executing some actions - for example printing labels. There are three types of events: file, serial port, database, TCP/IP server and HTTP server events.

File trigger event occurs when a specified trigger file is modified or created; Serial port trigger event occurs when data is received on the monitored RS232 serial port. Database trigger event occurs when a change in the monitored database table is detected. TCP/IP and HTTP server trigger event occurs when data is received on the monitored socket (IP address and port number).

**Trigger file** NiceLabel Automation Pro monitors the modification time and date of the trigger file and if the time of the trigger file is newer than the time the action was last executed, it executes the action. Changing the time and date of the file is called 'touching the file'. One of the ways to do that is using the MS-DOS command: REM>trigger\_file.

**XML** A metalanguage with the acronym, eXtensible Markup Language, that allows one to design a markup language, used to allow for the easy interchange of documents on the World Wide Web.