

Middleware Integration of Label Printing Into Existing IT Systems

Non-programming Integration White Paper

Version 20080606-01

Table of Contents

Table of Contents	2
1 Executive Summary	3
2 Middleware-Based Integration	4
2.1 Triggers	6
2.2 Mapping Data Filters	7
2.3 Trigger Actions	11
2.4 Using the Command Files	12
2.4.1 JOB files	13
2.4.2 XML files	13
2.4.3 CSV files	13
3 Conclusion	15
Appendix	16

1 Executive Summary

Labeling system integration, integration of new functionalities into existing IT systems, interconnectivity, interoperability... Once the technical terms are peeled away, a surprisingly simple solution is revealed. A middleware software module that "sits" between the labeling solution and your IT infrastructure, which makes sure they speak the same language.

This white paper focuses on the middleware method of printing functionality integration, a method that provides unique benefits in its application. It is the solution of choice for:

- Systems which cannot be reprogrammed to include a labeling solution or where the number of software platforms/dispersion of the system makes this ineffective
- Systems which evolve dynamically, making continuous programming-alterations resource-consuming and ineffective

At first glance, when compared with the other two methods, the middleware provides a way to fully integrate the printing functionality into your existing IT structure without modifying the structure that is already in place. It does not demand heavy modification of the programs that you are currently using, nor does it demand exporting and transport of printing files between NiceLabel and your system.

The non-programming integration described in this paper is based on **NiceWatch middleware**, a cutting-edge solution available in the labeling software market. Middleware solutions (to various degrees) feature powerful and flexible systems of data collection from many different sources and in just about every possible format, so that they can effortlessly use the communications from your current IT system and process this information in a way that the printing infrastructure understands.

The core benefits that non-programming labeling integration solutions offer are:

- The highest possible level of flexibility, due to a heavily customizable middleware module that easily adapts to changes in the existing system
- Complete cross-platform compatibility, as the middleware is able to gather data from any operating system
- Minimal resource commitment, the middleware module is configured and maintained easily and requires no programming knowledge (advanced feature flexibility is available with the help of VBScript customization).

2 Middleware-Based Integration

In many organizations, ERP systems run on mainframe computers that may or may not use a Windows operating system. Furthermore, integrating label printing into ERP systems by changing the source code might be very expensive to implement or simply not possible. Finally, enterprises may have their facilities located all over the world and use TCP/IP communication and email to send data and print labels from a centralized location to printers across the global network, or the databases may be protected to the point where software is unable to directly access them, but can process exported information from them.

This is where the middleware becomes the key player in connecting the existing system with the printing process, reaching out to both ends and handling the flow of information between seemingly incompatible systems.

NiceWatch provides a true real-time, automated label printing solution for any IT environment by performing the following:

- It gathers data from a number of sources and communication channels used by the existing infrastructure, reacting immediately when any change in data occurs
- This data is filtered, and pertinent printing information is extracted
- The extracted information is used by the actions defined in the trigger to either include the data in the label or its variable fields and immediately initiate printing by the print engine, or to further process the data using VBScripts

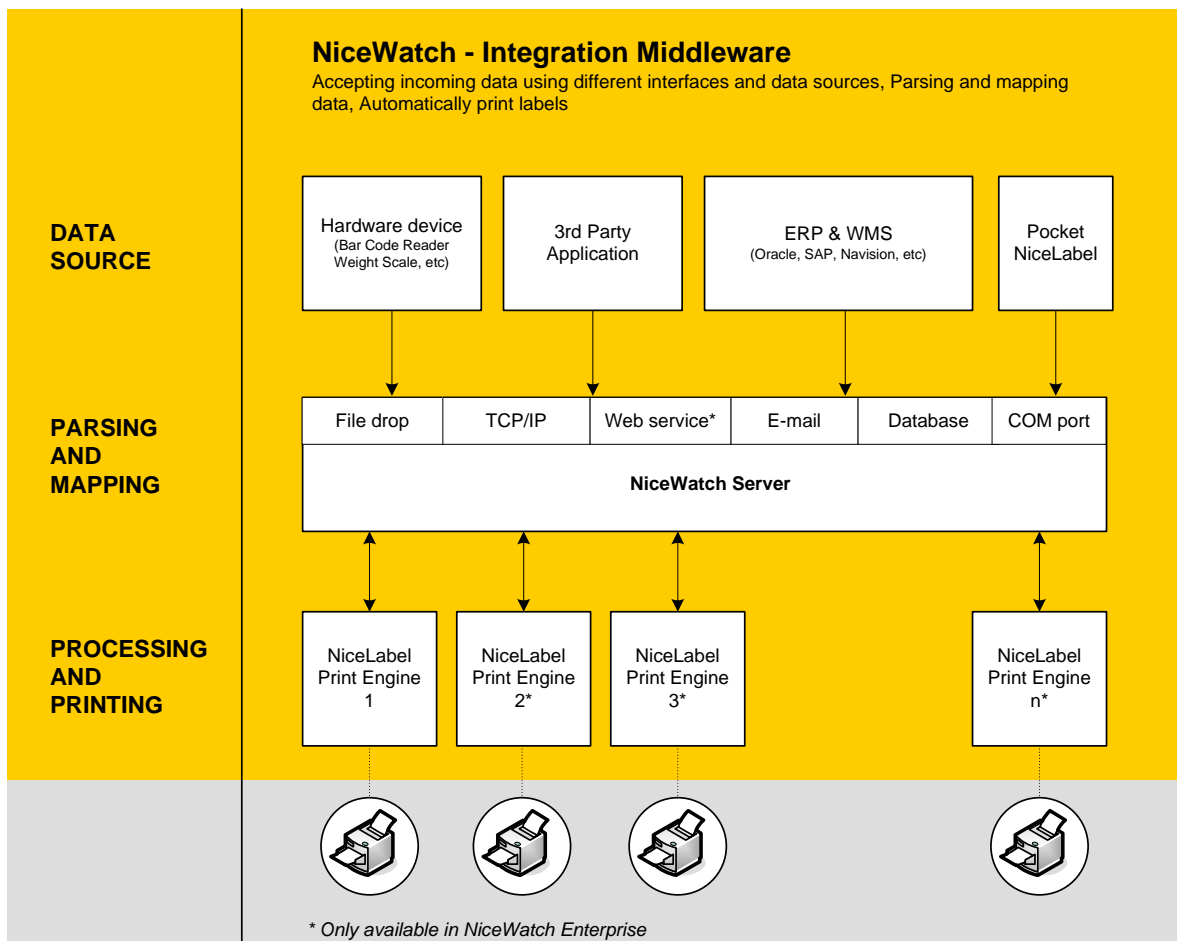


Figure 1: NiceWatch data workflow

The NiceWatch middleware tool extracts information from existing data channels and inserts it into labels, then initiates printing on virtually any type of printer. The core steps of the workflow are as follows:

- Event Triggers (grab data from existing applications or data channels immediately when it appears)
- Mapping Data Filters (map data fields with label variables, extracting pieces of relevant information from raw system data and connecting them with the appropriate variable label fields)
- Trigger Actions (execute printing or other commands)

2.1 Triggers

NiceWatch is an event-based application which monitors different system events. The monitoring activity is performed with the help of triggers, which are used to determine how and where NiceWatch monitors the system for data and event changes.

A trigger may monitor a file, a database, an e-mail account on a remote server, or a communication channel that connected devices use, be it physical or through the Internet. A trigger is poised to react to a change in the current condition of the monitored channel or location (although it can also be manually executed), and once such a change takes place, the trigger performs actions which have been assigned to it.

As an element of the complete IT structure, NiceWatch stands between your system and the print engine. The data monitoring method enables it to collect massive amounts of data through many channels and extract the key values from the often differently structured data formats, and present this work to the print engine. All of this is performed instantaneously and concurrently while the changes are made to the system by your existing IT infrastructure, makes NiceWatch a true tool of seamless integration.

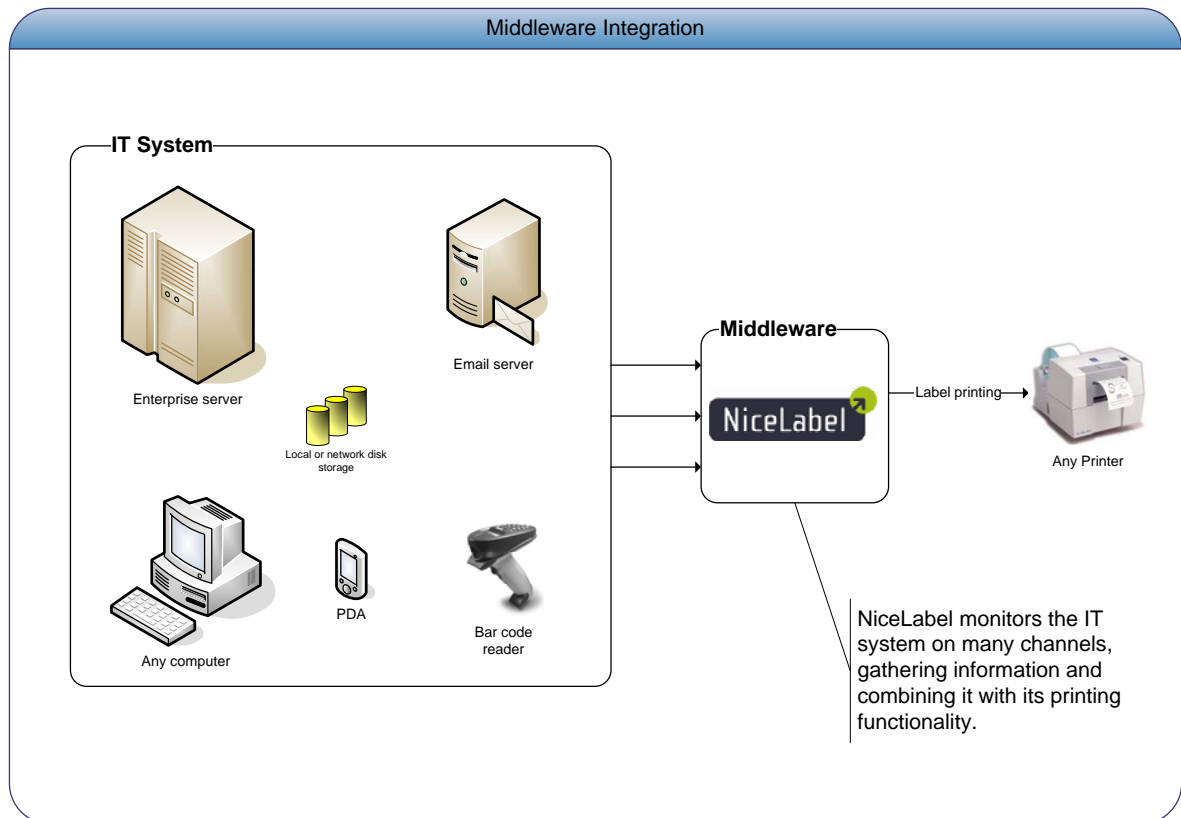


Figure 2: NiceWatch workflow

NiceWatch offers five (5) trigger options and NiceWatch Server (Enterprise edition) offers one additional trigger:

File Trigger:

A file trigger event occurs when a monitored file or a file in a monitored folder changes (the file modification timestamp changes). For example, the ERP application on the server exports the part of a database that holds the required label data. Those data fields are stored in a text file on a shared disk. NiceWatch detects the new file or the change of the file and initiates label printing. Variable values are extracted from predefined sections of the text file and transferred to the label where they are used at print-time.

- COM Port Trigger:** NiceWatch captures data received from a COM (serial) port and triggers an action. NiceWatch can receive different formats of incoming data. For example, a bar code scanner attached to the serial port scans the bar code for data. NiceWatch accepts the input data and initiates label printing.
- Email Trigger:** NiceWatch checks an email on a POP3 mail server. When an email is received, an email event is triggered and pre-defined actions are executed. The subject of the email message determines if the event should be triggered or not. The body of the message can hold the data for variable fields on the label.
- TCP/IP Trigger:** The command for starting label printing or the label data itself can be received over any specified TCP/IP port number. Mobile devices that are performing real-time data collection can send label requests to NiceWatch over the same wireless infrastructure and NiceWatch will print the labels to any printer on your enterprise network.
- The TCP/IP socket trigger enables mobile wireless data collection devices or any applications to send label requests to NiceWatch. Applications do not need to use FTP or Telnet to drop a file to a Windows server in order to trigger a label printing event.
- Database Trigger:** The database trigger checks a database for new or updated records in user-defined intervals. The trigger is a highly flexible, which allows it to address most of the various procedures in which database records can be handled.
- The database trigger offers a high level of integration and flexibility, as it seamlessly interacts with any systems which use (or are able to use) databases for data handling.
- Web Service Trigger:** A Web service trigger allows NiceWatch to monitor ports for Web-service communication. This offers a high grade of standardization in the communication channel, and the inherent benefits of Web service communication. The most applicable benefits in this case is the capacity to transfer data and the bidirectional communication channel, which allows for confirmation checking.
- (Enterprise Series only)**

2.2 Mapping Data Filters

Because the channels of data collection with triggers are extremely varied, and because NiceWatch is designed to be able to integrate into virtually any type of ERP or other existing IT infrastructure, the software does not require received data to be structured in a certain way.

Instead, it provides advanced and powerful options of data mapping, allowing it to adapt to virtually any type of data, whether it's data from a database, text file, XML database, or any other data.

NiceWatch parses input data, extracts required data fields as set in the criteria, and maps them to the variables defined in the label. If the data you have received from your trigger cannot be used immediately in NiceWatch, you have to use filter options in NiceWatch to extract only the portions of the data that you need to print the labels.

NiceWatch offers three (4) filter types to process data in unstructured and structured incoming data:

- Document analysis of unstructured text files
- Text database analysis of structured text files
- XML database analysis of XML files
- Database analysis

Document analysis

Parses an unstructured text file for data that you want to print on your label. **Unstructured text files** are for example reports, invoices or printer data stream files. 'Unstructured' refers to unformatted data structure with data not organized in a table-like format. The data is not organized in rows and columns that are separated with a separator character such as a comma, or have fixed column width. NiceWatch can extract specific data items within an unstructured file that you want to assign to certain variables.

The invoice below shows the highlighted data that you want to print on your label. Usually, you extract the lines with the data and then use the data fields individually on the label.

```

Invoice
One Portals Way, Twin Points WA 98156
Phone: 1-206-555-4756

Date: 20-feb-2005

Ship To:                               Bill To:
SPAR Supermarkt                        Richter Supermarkt
Starenweg 5                             Grenzacherweg 237
Geneve 1204                              Geneve 1203
Switzerland                              Switzerland

Order Customer Salesperson Order Date Required Date Ship Via
11075 RICSU Laura Callahan 06-may-1998 03-jun-1998 United Package

Product ID Product Name Quantity Unit Price Discount Extended
-----
      2      Chang           1      €190,00      15%      €161,50
     46     Spegesild           1      €120,00      15%      €102,00
     76    Lakkalikoori          2       €18,00      15%       €30,60
    
```

Figure 3: Unstructured Data Text File

The column 'Quantity' in the invoice above determines the amount of labels that NiceWatch will trigger to print (a total of four (4) labels). See figure below.

Report Label		Report Label	
Ship To: SPAR Supermarkt	Bill To: Richter Supermarkt	Ship To: SPAR Supermarkt	Bill To: Richter Supermarkt
ID: 2	Product name: Chang	ID: 46	Product name: Spegesild
Quantity: 1	Price: €190,00	Quantity: 1	Price: €120,00
Total: €349,11		Total: €349,11	
NiceWatch Sample Printed from NiceLabel Software.		NiceWatch Sample Printed from NiceLabel Software.	

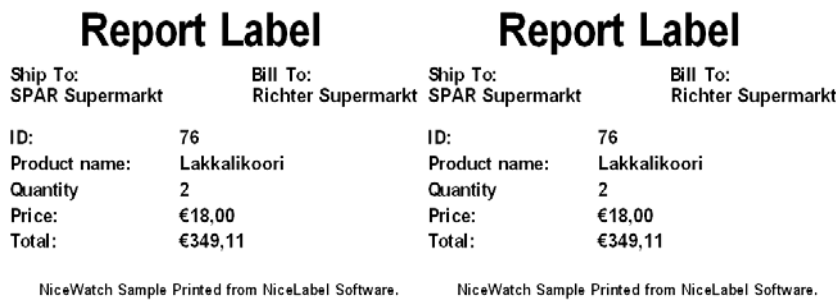


Figure 4: Labels with inserted filtered data

The figure below shows another type of unstructured document (printer stream file) that you parse with the Document analysis filter. NiceWatch extracts the highlighted data fields and maps them to the variables on the label. Data fields from one printer's data stream can be printed on another printer, making a converter between print streams of different printer brands a possibility.

```

^XA~TA000~JSO^LT0^MMT^MNW^MTT^PON^PMN^LH0,0^JMA^PR6,6^MD0^JUS^LRN^CI0^XZ
^XA^LL0480
^PW719
^FT399,171^A0N,28,28^FH\^FDRichter Supermarkt^FS
^FT52,171^A0N,28,28^FH\^FDSPAR Supermarkt^FS
^FT294,231^A0N,28,28^FH\^FD2^FS
^FT294,268^A0N,28,28^FH\^FDChang^FS
^FT294,305^A0N,28,28^FH\^FD1^FS
^FT295,380^A0N,28,28^FH\^FD€349,10^FS
^FT294,343^A0N,28,28^FH\^FD€190,00^FS
^FT152,89^A0N,68,67^FH\^FDReport Label^FS
^FT397,140^A0N,28,28^FH\^FDBill To:^FS
^FT52,140^A0N,28,28^FH\^FDShip To:^FS
^FT52,231^A0N,28,28^FH\^FDID:^FS
^FT52,268^A0N,28,28^FH\^FDProduct name:^FS
^FT52,305^A0N,28,28^FH\^FDQuantity^FS
^FT94,448^A0N,23,24^FH\^FDNiceWatch Sample Printed from NiceLabel Software.^FS
^FT52,343^A0N,28,28^FH\^FDPrice:^FS
    
```

Figure 5: Unstructured File: Print Stream Data File

The parsing and mapping filter technology works with any type of unstructured text file that you wish to use as a source of variable data on your labels.

Text database analysis

Parses a structured text file for data that you want to print on your label. **Structured text files** are text files with label data, which can include optional label and printer information. ‘Structured’ refers to formatted data structure with data organized in a table-like format, using a fixed width of data fields or data fields separated with a certain character. See the sample figures below.

The samples below show a text database with fixed length of the fields that are formatted in columns and a text database using commas to separate the data fields. Each line in the file contains a record with data fields for a single label. Each column provides data for one variable on the label.

1161890-03	1161890	378	67871130	HALTER	Test,200,2
1162008-03	1162008	884	67871350	FUSSRASTENPL	Software,101,1

1162132-03 1162132 150 67871125 BREMSHEBEL	Printing,55,3
1162212-03 1162212 105 67871134 LAGERBOCK	

Figure 6: Structured File: ASCII Text Database

Figure 7: Structured File: Comma-Separated Data Fields

XML database analysis

Parses an XML file for data that you want to print on your label. The sample below shows a generic XML database file setup. The XML filter will map data values from incoming XML files to variable fields defined in the label. The XML structure provides the elements, attributes and the name of the variables that are printed on the label.

Additional formats of XML databases are supported, Oracle XML, which is the standard output of the Oracle WMS applications, and NiceLabel XML. These, however, have built in support in the shape of actions and do not need to use filters to extract information.

```
<entry>
  <name>John Doe</name>
  <email href="mailto:jdoe@emailaholic.com"/>
  <address>
    <street>34 Fountain Square Plaza</street>
    <region>OH</region>
    <postal-code>45201X</postal-code>
    <locality>Cincinnati</locality>
    <country>US</country>
  </address>
  <tel preferred="513-555-7098">513-555-8889</tel>
</entry>
```

Figure 8: Structured file - generic XML file

Database analysis

This filter is only available for the Database trigger, and it parses a structured database for data. As the trigger already implies, the data is collected from a supported database, and the filter merely needs to extract the fields and the information within them.

As the recognized database is divided into fields, there is no need for advanced field filtering. Instead, the filter collects the fields from the database, and allows the user to connect each of those fields with a variable from the label.

For further information, see the *NiceWatch User Guide* available for download from the NiceLabel Web site at www.nicelabel.com.

2.3 Trigger Actions

In much the same extent that triggers monitor and gather information, actions handle the information once it has been collected. When an event occurs (file drop, COM port, email or TCP/IP communication, database change, web service broadcast), the trigger actions tell NiceWatch what to do. Actions include commands such as:

- 'open the label'
- 'set the variable on the label to a certain value'
- 'select printer for printing'
- 'print the required amount of labels'

The actions are executed in a sequence, which allows for a predictable flow of information. Also, each action can be limited by conditions, which it must fulfill in order to be executed.

The information has entered NiceWatch from the ERP, and actions are the tool which tells NiceWatch what to do with it. In the case of label printing, NiceWatch runs the first action in the list, and works its way down through what could be a fairly complex system of actions sequentially. It can open a label, set which printer to use for that particular trigger, print the label with the inserted variable values, run command files, send information to a COM port, or employ whichever commands the user wishes to program with the help of VBScript.

The options again are broad and many, to the extent where NiceWatch can be used for a spectrum of dynamic data collection and processing tasks, not even necessarily connected with printing. For example, the flexibility of actions allows the user to extract data from a new file or an e-mail message, and then automatically launch a pre-determined program, using the extracted data as a command line parameter during the launch.

If an action cannot be executed, NiceWatch records an error with a description into a log file that helps to identify and resolve the problem.

Once the actions are executed, assuming that the physical printing infrastructure is in order, the commands are executed and the labels are printed. NiceWatch Enterprise is intended for the enterprise user, where in a potentially global networked environment, an error may be harder to notice and could carry serious consequences. This is why the NiceLabel Enterprise Edition offers a vital central printing management tool. The Enterprise Print Manager provides unparalleled oversight and control functions for the complete printing process, as it centrally collects all printer status, printing errors, and provides the user with powerful process optimization tools. It is also capable of alerting the user instantly as an error occurs, allowing for the shortest possible reaction time and minimizing the impact of the reported error on the work process.

2.4 Using the Command Files

Command files are text files which enable the user to instruct the print engine on what to do. They use the NiceCommands structure in order to send the commands to the print engine.

The following three command file types are supported:

- JOB file
- XML file
- CSV file

The command files can be used by NiceWatch through the Run Command File action, and they provide a powerful means of unattended printing. They essentially perform a sequential string of commands, for example instructing the print engine open a label file, setting the values of variables on the label, printing a set amount of labels on a specific printer in the IT system, and similar.

2.4.1 JOB files

NiceCommands are usually used in JOB files printed from NiceWatch. A .JOB file is a simple text document containing NiceCommands.

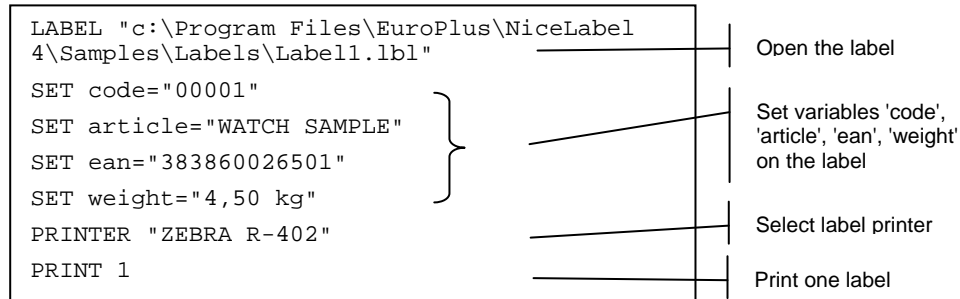


Figure 9: JOB file containing NiceCommands

2.4.2 XML files

The commands available in the XML command files are a subset of NiceCommands. The syntax of the commands differs somewhat, and there is a prescribed command file structure, in order for the print engine to understand the commands.



Figure 10: XML command file

2.4.3 CSV files

The CSV stands for Comma Separated Values. This command file is a text file, where individual field values are delimited by the comma (,) character. The first row must contain field names, and each subsequent row in a CSV file contains the command for one set of label printing.

```
@Label,@Printer,@Quantity,@Skip,ShipTo,Address,City,Country,ZIPCode,CompanyNumber,Counter  
C:\label\CmdFile.lbl,HP LaserJet 2300 Series PCL 6,1,1,Felipe Hernandez,122  
Jump St.,Barcelona,Spain,34938,65791,1  
C:\label\CmdFile.lbl,ZEBRA 140XiIII Plus,2,2,Yvonne Marseille,965 Rodeo  
Dr.,Paris,France,48025,04687,10
```

Figure 11: CSV command file

3 Conclusion

Inevitably, this document focused on non-programming labeling integration through the lens of the NiceWatch software module. While the specifics of individual settings and approaches may differ between the market-leading middleware modules, the core methodology and the approach to the problem is, for the most part, identical.

Middleware modules, with varying degrees of success, are positioned between existing infrastructures and labeling solutions. They reach out into the various areas of the infrastructure and siphon printing-relevant information as it appears. The information is processed, the strings of data that are needed for labeling are pulled out, and inserted into the label, and the label printing is initiated.

Any system which is either too fragmented between geographical locations, operating system platforms, or bound to constant change and evolution has no real option of using a programming solution. Systems that regularly evolve, change, and adjust to a dynamic business environment also can't apply programming in a way that can justify the needed programming resources and adjust in a timely manner.

Such systems optimally apply the highly flexible method of middleware integration, a method in which the integrated solution works seamlessly expands the functionality of your system and does not slow its dynamic development and evolution.

If you want to learn more about the different label printing integration methods or the NiceLabel product range in general, see the literature referenced in the Appendix of this white paper or visit the NiceLabel Web site at www.nicelabel.com.

Appendix

Additional Resources

To learn more about NiceLabel or how to integrate bar code and RFID label design and printing into your IT system, see the following list of additional resources and documentation. The documents are available at <http://www.nicelabel.com/Learning-center>.

General NiceLabel resources	<ul style="list-style-type: none">▪ NiceLabel Web site Learning Center▪ NiceLabel Tutorials▪ NiceLabel Technical FAQ▪ NiceLabel Technical Support site▪ NiceLabel forums
Middleware Integration:	<ul style="list-style-type: none">▪ White Paper: The Concerns and Benefits of Label Printing Integration▪ NiceWatch User Guide▪ NiceWatch Offline Help
NiceLabel Pro:	<ul style="list-style-type: none">▪ NiceLabel Quick Start Guide▪ NiceLabel Pro User Guide▪ White Paper: Multi-lingual label printing with Unicode▪ White Paper: NiceLabel RFID Technology

Glossary

Event Trigger	NiceWatch is an event-based application for label printing that monitors different system events. If a file drop, COM port, email or TCP/IP triggers an event, predefined actions will take place, Actions can be 'open label,' 'set variable,' 'set printer' and 'print label.' Event triggers and trigger actions enable NiceWatch to perform true real-time, on-demand and automated label printing.
NiceCommands	Simple text commands that enable third party applications to control automatic label printing with NiceLabel. A .JOB file is a text document that contains NiceCommands.