Copyright© 2005 Information Pathways Inc.



iBars™ User Manual

Beta Version 1.0

February 15, 2005

Table of Contents

Introduction	.3
How to Use This Document	.4
Setup Overview	.5
Installation	.5
System Requirements	.5
Setting the Search Path	.5
Running iBars TM	.5
Design Time	. 5
Run Time	.6
The XML File	.6
LABEL	. 7
PRINTERNAME	.7
QUANTITY	.7
JOBNAME	. 8
Variables	.8
Template Files	. 8
iBars [™] XML Mapper	.9
Installation	.9
Running the iBars [™] XML Mapper	.9
iBars™ XML Parser	10
Installation1	10
Running the iBars [™] XML Parser1	10
Command Line Interface	10
Programmatic Interface	10
Synchronous Interface 1	11
Socket Interface	11
Asynchronous1	11
iBars [™] Spooler1	12
Installation1	12
Running the iBars [™] Spooler	12
Appendix Q&A1	13

Introduction

The iBars[™] products have been developed for companies needing better run-time support for their current barcode label printing and who have a requirement to print bar code labels directly from ERP, MRP or in-house applications. Specifically, these products are designed to XML enable your existing printers and barcode label templates. iBars[™] products work with Oracle Warehouse Management System (WMS) and Mobil Supply Chain Applications (MSCA), as well as directly from custom applications.

The iBarsTM XML Mapper is specifically designed to work with existing customer templates that contain variable names delimited by special characters. The mapper then maps these variable names to the XML tag or variable names and creates a new run-time template. The iBarsTM XML Parser uses the new mapped template to merge run time data received in the form of an XML buffer to print the barcode label. The iBarsTM Spooler can be used to pick up XML barcode print files and send these to the XML parser, if an asynchronous and easy to set up solution is desired.

No new expensive printers need to be purchased. No new label design software needs to be purchased or learned. If you already have barcode label templates and you need to print from Oracle applications such as WMS or MSCA, this is the fastest, cheapest and easiest way to accomplish your goal. An added benefit is that you can simplify and standardize the interfaces of your in-house programs to print barcodes. Your programs just need to create a simple, standard XML layout and either write it as a file or send the XML as a buffer to the iBars[™] programmatic interface.

There are three iBars[™] programs described in this manual:

- iBarsTM XML Mapper
- iBars[™] XML Parser
- iBars[™] Spooler

How to Use This Document

The section **Setup Overview** provides general information concerning system requirements, installation and running the programs. This may be all you need. Additional information about installing and running each of the programs can be found under their respective headings.

Find the sections regarding DTD and XML to learn more about the XML that these programs are designed to handle. Finally, as an Appendix you will find Q&A. Hopefully, we will have anticipated your question. If not, please send us an email at <u>info@informationpathways.com</u>.

Setup Overview

Installation

iBars[™] programs are packaged as a java jar files and are installed simply by copying the small collection of files to the directory where you would like the executable to reside. The PATH variables may need to be set as per the specific operating system and shell used to execute iBars[™] from the command line. In Windows 98 and XP, if you have installed J2SE, there should already be an association between *javaw* and java archive files with a ".jar" suffix. The iBars[™] Mapper can be started simply by double clicking on the *mapper.jar* file once this association is established.

System Requirements

Version 1.4.2 of the Java Runtime Environment is required to run iBarsTM. iBarsTM has been tested on Linux, Windows 98 and Microsoft XP. While the design time portion of mapping the old label templates with the XML tags might be done in any environment supporting J2SE 1.4.2 or later which can be found at: <u>http://java.sun.com/j2se/.</u> It is recommended that the iBarsTM Spooler run in your production environment on a Linux or a Unix system for best results.

Setting the Search Path

If you install iBarsTM in a directory that is already in your search path, you should be able to start the program from the command line (See starting program), otherwise, you will need to add the directory where you installed iBarsTM to your search path. This varies as to operating system and if UNIX or Linux, which shell is used, but it may look similar to the following:

#Set iBars[™] environment variable for files residing in /programs/iBars[™] PATH=\$PATH:/programs/iBars[™] #Export the PATH if these lines are included in a profile export PATH

Please consult your system manual or man pages on setting the PATH variable for your shell or execution environment.

Running iBars™

There are two phases: 1) Design time and 2) run time. Design time involves mapping your template variables with XML tags and saving a template. Run time involves capturing the XML

files or buffers, parsing the data and merging the data with the template created during design tim.

Design Time

For label mapping you will need:

- Your template file
- Beginning and ending delimiters used to identify variables in your template file
- Sample an XML output file
- The name of your printer (if you are going to print a test label)
- the iBarsTM XML Mapper

Run Time

To test the run-time portion, you will need:

- The label template saved from the iBars[™] Label Mapper
- The name of your printer
- A sample XML barcode data file (may be the same one used in label mapping)
- To edit the XML data file so that _LABEL indicates the name of your label template and PRINTERNAME identifies the name of the printer on your test system
- the iBarsTM XML Parser

To prepare for run-time you will need:

- To coordinate the printer name and the value of the XML attribute _PRINTERNAME with your Oracle application administrator
- To coordinate the label template name and value of the XML attribute _FORMAT with your Oracle application administer
- The iBars[™] XML Spooler, or a program invoking the iBars[™] programmatic interface

The XML File

It will be helpful to review the layout of the XML file, if you are not already familiar with it. You may find you will need to edit the printername and label template attribute for testing purposes until these names have been coordinated with your application administrator. Ask your database or application administrator to create an XML barcode print file for you. Ask him or her to include all the tag names that they are likely to use for the type of label you are printing. You can also create this file manually, or add tag names to the file by using our example file.

<?xml version="1.0" encoding="ISO-8859-9" standalone="no"?>

<!DOCTYPE labels SYSTEM "label.dtd">

<labels _FORMAT ="label_template08" _QUANTITY="1" _PRINTERNAME="DJ720C" _JOBNAME="Test4"> label FORMAT="label template09" PRINTERNAME="ZebraS500">

<variable name= "ship_from_name">IPI</variable>

<variable name= "ship_to">Glen Dale Farms</variable>

<variable name= "from_addr">Traverse City, MI</variable>

<variable name= "ship to addr">Empire, MI</variable>

<variable name= "ship_from_CSZ">4</variable>

<variable name= "ship_to_CSZ">4</variable>

<variable name= "to_ZIP">49630</variable>

<variable name= "carrier">Fedex</variable>

<variable name= "ZIP bar">496841031</variable>

<variable name= "PO">1031</variable>

<variable name= "dpci">DPCI</variable>

<variable name= "case pack">Casepack</variable>

<variable name= "style">modern</variable>

<variable name= "color">blue</variable>

<variable name= "size">large</variable>

<variable name= "sscc18_hr">1234567890</variable>

<variable name= "sscc18_bar">1234567890</variable>

<variable name= "DESCRIPTION">Test Label</variable>

<variable name= "QUANTITY">82</variable>

</label>

</labels>

The first line of the file identifies the version number of XML and character set used. The second line identifies this file as a barcode data file following the syntax defined in the *label.dtd* file. The *labels* tag can define attribute values for the entire batch of labels. These attributes can be overridden by redefining the attribute for each separate label contained between the *<label>* and *</label>* tags.

_LABEL

The name of the template file, or "_LABEL" in the XML file tells us which label template to use during run time. These names need to be coordinated between the Oracle application administrator and the person responsible for barcode templates. The value associated with

_LABEL needs to be the name of the file that the new template mapping is saved as. The file names can be changed later, but it is easier if the correct name is assigned. It is likely that the

Oracle application administrator will leave the template naming up to the barcode labeling administrator.

PRINTERNAME

The _PRINTERNAME attribute tells the iBars[™] run time program which printer to send the output to. This should be the system name of the printer. This name needs to be associated with a printer during the run time setup procedure. It is mentioned here just for completeness.

_QUANTITY

Quantity is the number of copies of all the labels or each label to print, depending upon if it is set at the *labels* level or the *label* level.

_JOBNAME

Jobname may be optionally set to identify this particular job. It can be useful for identifying job status or in resolving printing issues.

Variables

Variables are identifyied with the "*name*=" syntax, followed by the value of the variable between the ">" and "<" signs. You may add additional variable names by adding a name after "*name*=". The actual data associated with the tag or variable resides between the ">" and "<" characters. The data field may be left blank for the mapping activity, but keep a null set, "><", so the XML file will parse correctly.

Template Files

You will need to tell the iBarsTM mapper what the beginning and ending delimiters are for your variables. These are the character or characters that are positioned immediate before and after the field where variable data is inserted in your template. Make a note of the beginning and ending character strings, as you will need to tell the iBarsTM mapper what these characters are.

iBars[™] XML Mapper

The iBars[™] XML Mapper works by mapping the variables in your old templates to the tags (variables) in the XML barcode print files. The XML Mapper program is a tool to facilitate the easy mapping of variables in your existing template with the XML tags or variables in the XML data file and put the result in a new template that can be processed during run time by the iBars[™] XML Parser.

Installation

The iBarsTM XML Mapper installation requires the following files:

- mapper.jar
- mapper.jarContent
- manual.htm
- key.sbj

Copy these files into a folder or directory on your system. There is also a sample template file with a ".frm" suffix and a sample XML data file for testing purposes. Your template file is not required to have any particular name or suffix. Simply place these files in a directory on your system.

Running the iBarsTM XML Mapper

The iBarsTM XML Mapper serves as an easy means to map the variable names in your legacy templates with the XML tag names associated with the ten Oracle label types. The mapping could be accomplished manually, but this step also creates a template that creates an association with the XML barcode label file during run time. The iBarsTM XML mapper should run on any system with the Java runtime environment of version 1.4.2 or later. Simply double click on the *mapper.jar* file. If there is no file association, associate it with *javaw*, which will recognize a jar file. For command line in windows, Unix or Linux, type:

java -jar mapper.jar

A window will open and a status bar internal window should be visible that will provide hints to map your first label template. You will need to open your template file, open a sample XML file that will have all the variables or tags that will be contained in your run-time labels, and you need to enter the beginning and end delimiters of the variables in your legacy template file. Then in the Tools menus perform the mapping. This will bring up a window showing your variable names on the left and a "spinner" control to select the XML tag that you want associated with that variable. Map all the variables and then select save and exit from the File menu. Lastly, save your new template to a file. Keep in mind that this file name will be the name passed via the XML barcode

file as the value of the _FORMAT attribute. This needs to be coordinated with the ERP application administrator, along with the printer name.

iBarsTM XML Parser

The iBarsTM XML Parser takes an XML barcode file that adheres to the convention of the Oracle XML Barcode DTD and prints barcodes in a format described by a template created by the iBarsTM Mapper program. The iBarsTM XML Parser contains a command line interface for testing, a java programmatic interface, as well as the specific interface defined by the Oracle Synchronous Interface.

Installation

The iBars[™] XML Parser installation requires the following files:

- parser.jar
- parser.jarContent
- key.sbj

Copy these files into a folder or directory on your system.

Running the iBarsTM XML Parser

Command Line Interface

There are a few ways to run the iBars[™] XML Parser. A convenient way to test your new label template and print your first label, is by passing a test XML file to the parser from the command line. This may be the same file you used to perform the mapping. Open a shell (in Unix or Linux) or DOS window in Windows and assuming your test file is called *test01.xml*, type the following:

java -jar parser.jar test01.xml

If your label did not print, edit the *test01.xml* file and make sure that _FORMAT is set to the value of your template file and that your template file is residing in the same folder or directory as the parser.jar executable. Also, check that _PRINTERNAME is set to the name of the printer. This name can be determined by finding your printer via the printer utility appropriate to your operating system and checking the name. With printer name and template file name set, you should print your first label. If not check the *eventlog.txt* file.

Programmatic Interface

The *parser.jar* file contains the entry points to pass a buffer to the parser representing an XML formatted barcode label print file. The package name is com.ipi.barcode.parser. The following import line must appears in your java program:

// reference the package where the iBarsTM XML parser resides

import com.ipi.barcode.parser;

The following is the interface to the program, assuming that *mystring* is of type java.lang.String and contains the XML buffer:

// create an instance of the parser

XmlParser xp = new XmlParser();

// pass the java.lang.String string containing the XML barcode label

xp.entry(mystring);

Synchronous Interface

The Oracle Synchronous connection is a programmatic interface as per the Oracle interface specification.

The package name is com.ipi.barcode.parser. The following import line must appears in your java program:

// reference the package where the iBars[™] XML parser resides

import com.ipi.barcode.parser;

The following is the interface to the program, assuming that *mystring* is of type java.lang.String and contains the XML buffer:

// create an instance of the parser

XmlParser xp = new XmlParser();

// pass the java.lang.String string containing the XML barcode label

XmlParser.syncPrintRequest(buffer,printer_status,job_status,status_type);

Socket Interface

TBA

Asynchronous

(See iBarsTM Spooler.)

iBarsTM Spooler

The iBars[™] Spooler is the easiest way to set up barcode label printing for your Oracle Warehouse Management System (WMS) and Mobil Supply Chain Applications (MSCA). A simple command line interface will run the iBars[™] Spooler as a background process that will continue to process labels defined by the iBars[™] XML Mapper, by sending the XML buffer to the iBars[™] XML Parser.

Once your legacy templates have been translated, barcode labels can be printed by using the iBarsTM XML Spooler or via a direct programmatic interface to the iBarsTM XML Parser. The iBarsTM XML Parser also supports the interface specified by Oracle for their Warehouse Management System (WMS) and Mobil Supply Chain Applications(MSCA).

Installation

The iBars[™] Spooler installation requires the following files:

- XmlSpooler.jar
- XmlSpooler.jarContent
- key.sbj

Copy these files into a folder or directory on your system.

Running the iBars[™] Spooler

The iBars[™] Spooler provides the easiest method to print barcode labels from your ERP application. The *spooler.jar* may be run on any system that has the Java run time environment installed and that has access to the folder or directory where the ERP application is placing the XML label files. The iBars[™] Spooler must have write access to this directory so that it can remove files that have been printed. It may be run in Windows by double clicking on the spooler.jar file, provided a recent version of the Java run time has been installed and java archive files are associated with the javaw application. Simply set the timer interval and locate the spool directory and start the spooler. The spooler my be stopped and restarted with the graphical interface.

Most Unix or Linux administrators will prefer to run the spooler as a background task or daemon from a script file. This can be easily accomplished by typing the following:

```
java -jar spooler.jar /usr/spool/erp 10 &
```

The first parameter is the full path of the spool directory */usr/spool/erp*, in this case. The second parameter is the number of seconds to delay before again checking the directory for more print files, *10* seconds in this example. Lower numbers will print your labels sooner and higher numbers will have less performance impact on the server. The ampersand should be omitted for

Windows systems, as this indicates to the Unix or Linux shell to run the program in the background.

Appendix Q&A

Q. How do I know this program will work for my template files?

A. If you have created variable names in a print file and created character delimiters to mark the beginning and end of variable names, this program should work for you.

Q. Why are there three programs?

A. We thought it wise to separate design time activities from run time. Often the people assigned to designing and managing labels are not the same people who print the labels. The spooler runs as a separate task and is only one way to print labels, so it is the third program.

Q. Do the attributes and variables need to be in the same order.

A. Attributes must appear immediately after the "*<label*" or "*<labels*" tags, but may appear in any order. Likewise, the variables are not order dependant. Please be aware that attributes that appear after the "*<labels*" tag will take precedent over the ones after the "*<label*" tag.

Q. Once an attribute is set after a "<label" tag, how long does it take effect.

A. Only until the end of that label, or until the ending "</label>" tag, after which point the attribute will revert back to the "<*labels*" setting.

Q. My label printed during design time testing, but did not print during run time. What could the problem be?

A. Check that the *_PRINTERNAME* attribute is assigned to a printer name that can be found on the run time system. Check that the *_FORMAT* attribute value matches the name of the label template that resides in the same directory as the parser.jar file.

Q. I am still not able to print. What should I do next?

A. Open the *eventlog.txt* file and look for error messages. This file should appear in your home directory, the execution directory, or in the spool directory. You should be able to tell if the printer or label template were not found, or if there were problems reading the XML file or your legacy template files. Your XML file should have a "*.xml*" suffix.

Q. I am still not able to print. What do I do now?

A. Please send your XML file, your legacy template file and your evenlog.txt file, along with an

explanation of your problem to info@informationpathways.com.

Q. The program won't run at all.

A. Make sure you have the *key.sbj* file in the execution directory for mapper.jar, parser.jar and spooler.jar. Make copies of the file if the jar files are in different directories. If your key file has expired, you may download another one from our web site.