



White Paper

# XML Integration with Oracle's WMS and MSCA

How to Automatically Print Labels from  
Oracle Applications

|   |    |
|---|----|
| <b>Introduction</b> .....   | 4  |
| <b>Overview</b> .....   | 4  |
| <b>Label Printing Integration with Oracle Warehouse Management and Oracle Mobile Supply Chain</b> ..... | 4  |
| <b>Seagull Scientific's Commander Utility</b> .....   | 4  |
| <b>Differences in Commander Editions</b> .....  | 5  |
| XML Capabilities with the Enterprise and RFID Enterprise Editions.....                                  | 5  |
| Enhanced Capabilities with the Enterprise Print Server Edition.....                                     | 5  |
| <b>Included Files</b> .....   | 6  |
| Label Formats .....   | 6  |
| Commander Task Lists .....  | 6  |
| XML2TXT Utility (for Enterprise and RFID Enterprise only) .....   | 7  |
| <b>Configuring Commander for Integration with Oracle WMS/MSCA</b> .....                                 | 7  |
| <b>Integration Steps</b> .....  | 7  |
| <b>Testing the Commander Configuration</b> .....  | 7  |
| <b>Finishing the Integration</b> .....  | 8  |
| <b>Appendix A: Enterprise Print Server Implementation</b> .....   | 8  |
| <b>The Two Tasks of OracleXML-PrintServer.tl</b> .....  | 8  |
| <b>Appendix B: Enterprise and RFID Enterprise Implementations</b> .....                                 | 8  |
| <b>How Commander Processes Oracle XML Print Requests</b> .....  | 9  |
| <b>Task #1: Converting from Oracle XML to Commander Script</b> .....                                    | 10 |
| Example of Converting from XML to Commander Script .....  | 10 |
| <b>Task #2: Executing Commander Script to Print Labels</b> .....  | 11 |
| <b>Processing Multiple Print Jobs within a Single XML File</b> .....                                    | 11 |
| <b>Mapping the _FORMAT Attribute to a BarTender Format File</b> .....                                   | 11 |
| <b>Appendix C: XML2TXT Command Line Parameters</b> .....  | 12 |
| <b>Parameter Reference</b> .....  | 12 |
| <b>Command Line Examples</b> .....  | 13 |
| Explicit XML File Name .....  | 13 |
| Implied XML File Name.....  | 13 |
| Command Line Example: Diagnostic Arguments .....  | 13 |
| <b>Appendix D: Configuring XML2TXT</b> .....  | 14 |
| <b>Format Map</b> .....   | 14 |

|   |           |
|---|-----------|
| <b>Custom Commander Script Arguments.....</b>     | <b>14</b> |
| <b>Appendix E: Troubleshooting XML2TXT .....</b>  | <b>15</b> |
| <b>Log File .....</b>                             | <b>15</b> |
| <b>Error Dialogs.....</b>                         | <b>15</b> |
| <b>Running XML2TXT from a Network Share .....</b> | <b>15</b> |
| <b>Appendix F: For More Information .....</b>     | <b>16</b> |

## Introduction

This white paper explains how Oracle's WMS and MSCA can be integrated with BarTender to automatically print labels. This very simple process is covered in the first few pages. Appendices A-E provide additional technical details that are not required knowledge to do the integration.

## Overview

### ***Label Printing Integration with Oracle Warehouse Management and Oracle Mobile Supply Chain***

Oracle Warehouse Management (WMS) and Oracle Mobile Supply Chain (MSCA) are logistics management software products within the Oracle E-Business Suite. Both WMS and MSCA offer integration capabilities that allow for automatic label printing when used with the right external software products, such as BarTender from Seagull Scientific.

There are two primary tasks that must be completed in order to implement this label printing integration:

- **Configuring Oracle's WMS and MSCA Applications.** Label printing integration with Oracle's WMS and MSCA is based on the generation of an XML file (called an "XML label request") containing all of the information needed to describe a label job. These applications must be configured to create the desired XML file and place it in the proper folder. The steps to do this are not documented here but can be found in the "WMS Implementation Guide," available online to Oracle partners and customers from <http://metalink.oracle.com>. Oracle is responsible for helping users get to the point where they are properly generating these XML files. For additional information on Oracle WMS/MSCA and label integration, please see [Appendix F](#).
- **Configuring Commander and BarTender.** To handle XML label requests, Commander must be configured both to monitor the folder in which the XML files will be created and then respond by executing the desired print job. This white paper documents the steps necessary to accomplish that.

### ***Seagull Scientific's Commander Utility***

Commander is a utility provided with all BarTender Enterprise editions that allows BarTender to automatically print label jobs in response to certain triggering events from other software. One of these events can be the generation of a "trigger file" (such as an XML label request) in a particular file folder. When a file of a given name or type is detected, Commander reads commands and/or data from that file and passes them on to BarTender, which executes the print job. Commander can also be triggered by and process data sent over a TCP/IP socket.

For more extensive information about using the Commander utility, please see the Commander white papers at:

<http://www.seagullscientific.com/aspx/whitepapers.aspx>

## ***Differences in Commander Editions***

The Enterprise Print Server edition of Commander provides some functionality that is not available with the RFID Enterprise and base Enterprise editions.

### ***XML Capabilities with the Enterprise and RFID Enterprise Editions***

Although Commander can detect the creation of Oracle-generated XML files, it does not understand the XML contained in these files. The Enterprise and RFID Enterprise Editions of Commander therefore use another Seagull utility called XML2TXT to convert from Oracle XML into a file that contains Commander script (which Commander understands) and delimited label data. XML2TXT then creates a new trigger file for Commander. When Commander detects this newly-created trigger file, it reads the converted commands and the label data contained in the file and then tells BarTender to print the requested labels.

XML2TXT is easy to control from Commander using command line parameters, which makes the XML conversion process work like a built-in function of Commander. However, in contrast, the Enterprise Print Server edition of Commander (described below) does not require the use of XML2TXT.

### ***Enhanced Capabilities with the Enterprise Print Server Edition***

In addition to detecting file and e-mail triggers, such as that for Oracle XML print requests, the Enterprise Print Server edition of Commander can also trigger based on TCP/IP socket communications.

Also, the Enterprise Print Server edition supports transformation of incoming XML data into other formats using XSL. The XSL-based conversion process built into the Print Server Edition of Commander is a somewhat faster and more direct way to convert XML than is available in the Enterprise and RFID Enterprise editions of Commander, which executes the separate XML2TXT utility. For the special case of converting Oracle XML Print Requests into BarTender XML Script, Commander includes an XSL style sheet designed for this purpose.

Performance in heavy utilization environments is also improved with the Enterprise Print Server edition of Commander, as it can launch and communicate with multiple instances of BarTender.

## ***Included Files***

Several files are installed with the Enterprise editions of BarTender to make integration with Oracle applications easier.

## ***Label Formats***

When an Enterprise edition of BarTender installs Commander, it installs a set of label formats that implement Oracle's 10 "Seeded Label Formats". The format names and a brief description are listed below:

1. LPN (License plate information)
2. LPN Content (LPN plus details about contents)
3. LPN Summary (LPN plus nested contents summarized)
4. Material (Item, quantity, lot, and revision details)
5. Serial (Material plus serial details)
6. Location (Organization, subinventory, and locator)
7. Shipping (Customer and address)
8. Shipping Contents (Shipping plus content information)
9. WIP Contents (Discrete manufacturing details)
10. Flow Contents (Flow manufacturing plus customer information)

## ***Commander Task Lists***

Commander installs with two "Task Lists" dedicated to Oracle XML integration.

- **OracleXML.tl** is for use by the Enterprise and RFID Enterprise Editions.
- **OracleXML-PrintServer.tl** is for use by the Enterprise Print Server edition and provides the following advantages:
  - Supports reception of XML through a TCP/IP socket
  - Does not depend on the XML2TXT utility for XML conversion
  - Takes advantage of the Enterprise Print Server edition of Commander's support for multiple BarTender processes, which improve performance during heavy usage. (This task list is set to use two processes, but this number can be adjusted)

### ***XML2TXT Utility (for Enterprise and RFID Enterprise only)***

Although the XML2TXT Utility is installed with all editions of Commander, it is not needed by the Enterprise Print Server edition in order to convert XML. It is therefore only referenced in the OracleXML.tl task list mentioned above, but not OracleXML-PrintServer.tl.

## **Configuring Commander for Integration with Oracle WMS/MSCA**

### ***Integration Steps***

1. If using the Enterprise or RFID Enterprise editions, you will need to install the Microsoft .NET framework on your computer if it is not already installed. It is required by the XML2TXT utility. You can download the .NET framework from the Microsoft Download Center at:

<http://www.microsoft.com/downloads>

2. Install BarTender.
3. When BarTender installation is complete, use the Windows Explorer to browse to the Oracle samples folder, which should be located at “My Documents\BarTender\Formats\Samples\Oracle”.
4. You can find the “OracleXML.tl” and “OracleXML-PrintServer.tl” task lists in the Oracle samples folder. Open the one appropriate for your edition of Commander. Although both will work with the Enterprise Print Server edition, it is recommended that you use OracleXML-PrintServer.tl, as it will process the XML faster and can also accept XML through a TCP/IP Socket. (Attempting to load OracleXML-PrintServer.tl with the Enterprise or RFID Enterprise editions will result in an error message.)
5. Start detection in Commander. This should cause a folder named “Scan” to appear in the Oracle samples folder.

### ***Testing the Commander Configuration***

In order to confirm that you have Commander properly configured, perform the following test:

1. Select one of the sample XML files in the Oracle sample directory and copy this file into the “Scan” folder which appeared when you started detection in Commander.
2. BarTender should respond to the appearance of this XML file by automatically printing a label using data from the XML file.

## ***Finishing the Integration***

Once you have used this procedure to verify that Commander and BarTender are properly responding to the creation of XML files, all that remains is to configure the Oracle application to create the XML file in the Scan folder or send the data through a TCP/IP socket. You may also wish to use BarTender to load and edit some of the included Oracle “Seeded Label Formats” to better meet your specific requirements.

## **Appendix A: Enterprise Print Server Implementation**

It is not necessary to read these Appendices in order to complete a label printing integration with Oracle applications. However, they provide detailed technical information that may be of interest to some users. Appendix A applies to the Enterprise Print Server edition only. All other appendices apply only to the Enterprise and RFID Enterprise editions.

### ***The Two Tasks of OracleXML-PrintServer.tl***

There are two tasks in the OracleXML-PrintServer.tl task list: One that detects file triggers and one that is triggered by TCP/IP socket communications. Either method can be used with Enterprise Print Server.

Both tasks use an XSL style sheet to convert any Oracle XML Print Requests found in the trigger into BarTender XML Script. This script contains all of the BarTender commands and label data needed by BarTender to handle a print request. After generation by Commander, it is passed to BarTender’s Automation interface and executed. (Because the Print Server Edition supports XSL style sheet transformation, it does not need the XML2TXT utility.)

Both tasks contained in the OracleXML-PrintServer.tl task list share a single Command Handler that is configured to use two BarTender processes. This helps improve performance when handling multiple simultaneous print requests. If desired, use of additional BarTender processes can be specified.

## **Appendix B: Enterprise and RFID Enterprise Implementations**

It is not necessary to read these Appendices in order to complete a label printing integration with Oracle applications. However, they provide detailed technical information that may be of interest to some users.

This section takes a more in-depth look at how the Enterprise and RFID Enterprise Editions of Commander use the XML2TXT utility to generate a new Commander Script trigger file. The resulting second trigger causes Commander request a label print job from BarTender.



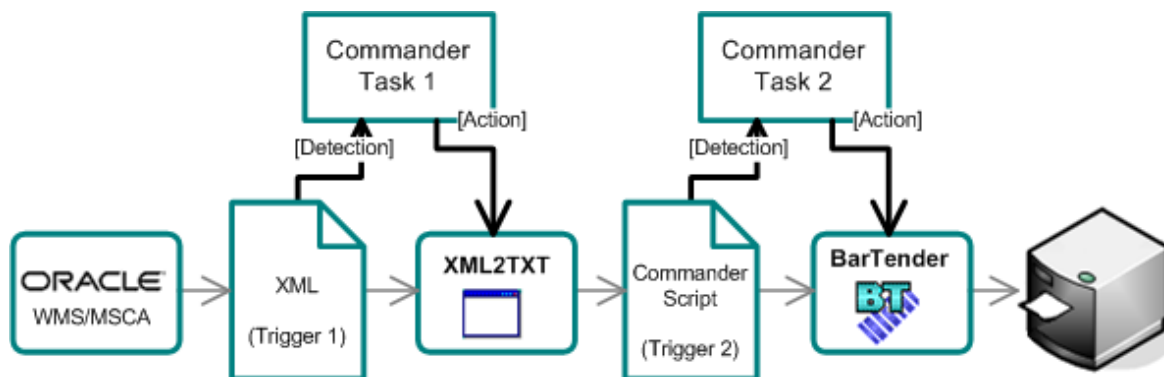
## How Commander Processes Oracle XML Print Requests

You can configure Commander to be “triggered” by new XML files and automatically run XML2TXT in response. XML2TXT will process the XML and produce another trigger file containing Commander script. Commander will then detect this second trigger file and pass the commands along to BarTender.

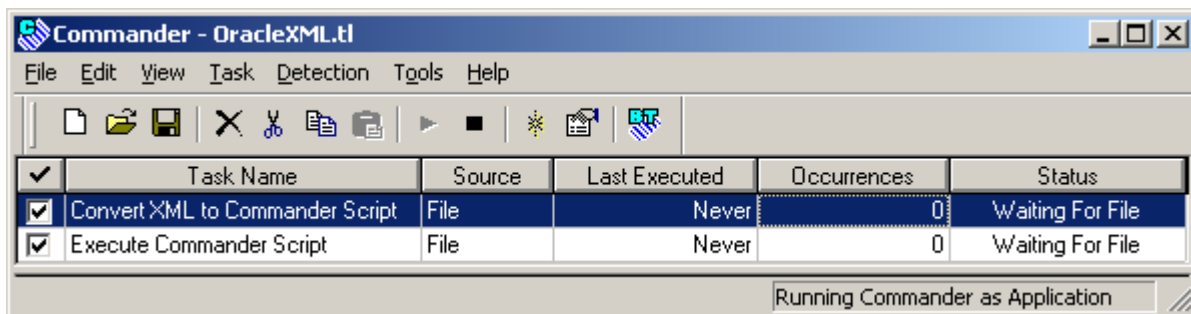
Commander works by managing a set of tasks that you define in a “task list.” Each task represents both a triggering event and commands to be executed by Commander in response to that event. Commander’s ability to simultaneously look for multiple trigger events allows you to define two tasks that run in sequence to perform the Oracle XML label integration:

**Task #1:** The first Commander task waits for a new Oracle-generated XML file. When a file with the extension “.xml” is detected, Commander runs XML2TXT in response. XML2TXT translates that XML into Commander script, which serves as a second trigger file.

**Task #2:** The second Commander task waits for the new trigger file containing Commander script. When this file is detected, Commander executes the script found within it, causing BarTender to print out the delimited data found in the trigger file.



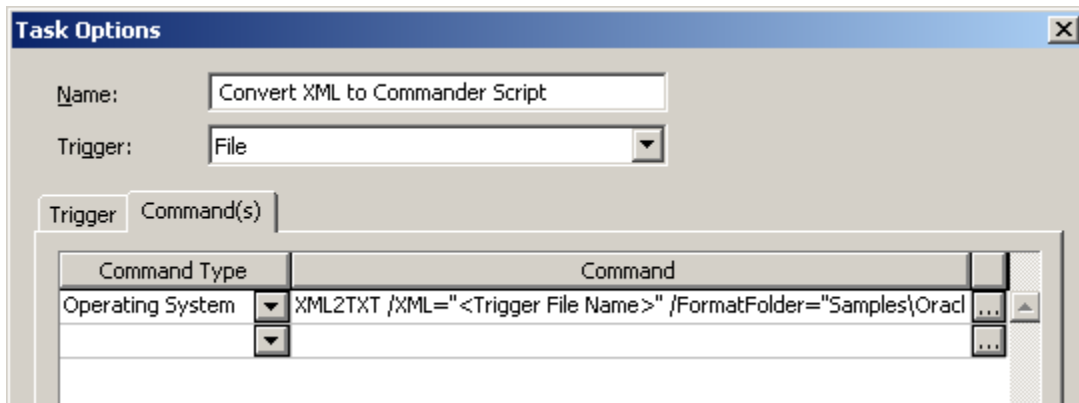
*Two Commander tasks work in sequence to print your Oracle XML labels.*



*How the two tasks might look in the Commander task list.*

## ***Task #1: Converting from Oracle XML to Commander Script***

The Commander task that converts Oracle XML to Commander script scans for XML files at the location where Oracle generates them. When Commander finds an XML file, it runs XML2TXT through a command line. You can define this behavior within your first Commander task by bringing up the Task Options dialog for that task in Commander, going to the Command(s) tab, and choosing “Operating System” for the Command Type, as shown below:



To the right of the Command Type appears the XML2TXT command line that Commander will execute. On the command line, you can pass in the filename of the XML file that will be converted into Commander script, or you can simply specify the Commander variable “<Trigger File Name>” and have Commander automatically fill in the correct XML file name. You can also specify the folder location of your Oracle label format files on the command line.

Here is an example of what the XML2TXT command line might look like in this Commander task:

```
XML2TXT /XML="<Trigger File Name>" /FormatFolder="C:\OracleFormats"
```

### ***Example of Converting from XML to Commander Script***

Here is an example XML file which might be generated by Oracle in order to print a license plate number label:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT ="Seeded Label Default for LPN" _QUANTITY="1"
  _PRINTERNAME="P1" _JOBNAME="LPN">
<label>
<variable name= "lpn">123ABC</variable>
<variable name= "organization">W1</variable>
</label>
</labels>
```

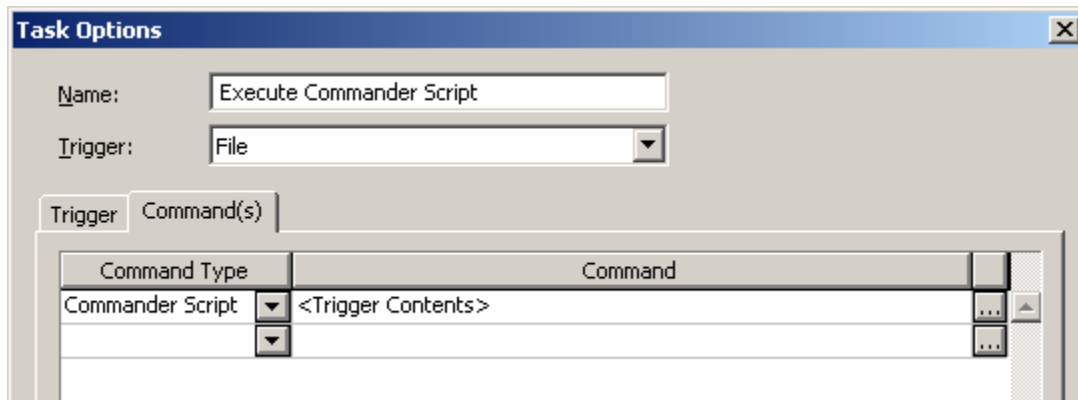
When Commander Task #1 gets “triggered” by the generation of this XML file, it calls XML2TXT, which produces a second trigger file containing Commander script and delimited data that might look like this:

```
%BTW% /AF="C:\MyFormatFolder\LPN.btw" /PRN="P1"  
/PrintJobName="LPN" /D="<Trigger File Name>" /C=1 /DbTextHeader=3  
/R=3 /P /DD  
%END%  
"lpn", "organization"  
"123ABC", "W1"
```

This second trigger file is processed by Commander Task #2 to print the labels using BarTender.

### ***Task #2: Executing Commander Script to Print Labels***

The second Commander task scans for Commander script files that have been converted from XML, executes the commands in these files, and then prints your Oracle XML labels using BarTender. You properly configure your second Commander task by bringing up the Task Options dialog for that task, going to the Command(s) tab, and choosing “Commander Script” for the Command Type, as shown below:



### ***Processing Multiple Print Jobs within a Single XML File***

Oracle’s XML syntax allows several print jobs to be requested from within a single XML label request file. XML2TXT handles this by creating a separate Commander script trigger file for each print job request contained in the XML trigger file. Commander will then automatically detect each Commander script trigger in turn and launch the appropriate BarTender label job for each one.

### ***Mapping the \_FORMAT Attribute to a BarTender Format File***

BarTender installs with a set of default label formats meeting the Oracle standards such as LPN.btw, Location.btw and FlowCont.btw. When XML2TXT reads in an XML file, it looks for an

XML attribute called “\_FORMAT”. Based on the value of this attribute, XML2TXT decides which label format file BarTender will print.

The “\_FORMAT” attribute can take either of two forms:

- **Explicit BarTender Label Format Name.** If the “\_FORMAT” attribute contains a .btw filename, BarTender will be instructed to open and print that label format. For example, the appearance of

```
_FORMAT="LPN.btw"
```

in the XML file will cause BarTender to open and print the LPN.btw format.

- **Seeded Oracle Format Name.** Alternatively, an Oracle format type may have been specified in the Oracle XML file instead of an actual BarTender label format file name. In this case, XML2TXT searches for the name of the appropriate BarTender label format to use in the Format Map section of the XML2TXT.config file. (For more details, please see Appendix C.) For example, if

```
_FORMAT = "Seeded Label Default for LPN"
```

appeared in the XML file, XML2TXT would look in the format map to determine that BarTender should use the LPN.btw format.

## Appendix C: XML2TXT Command Line Parameters

(This advanced information is for users of the RFID Enterprise and base Enterprise editions. The XML2TXT utility is not needed by the Enterprise Print Server edition.)

### Parameter Reference

```
/XML="[XML Filename]"
```

The full filename of the XML file to be processed by XML2TXT. When integrated with Commander, this argument will accept the Commander Variable “<Trigger File Name>” instead of the actual filename. Commander will then plug in the filename of the detected XML file before executing the command line. This argument is required.

```
/FormatFolder="[Path to Folder]"
```

The path which contains the seeded default label format files. If this argument is omitted or specifies an incomplete path, BarTender will attempt to find the label format automatically.

```
/Log="[Message Log Filename]"
```

The full filename of the XML2TXT message log. If not specified, logging is disabled.

**/Extension="[Trigger File Extension]"**

The extension used when creating the final trigger file from the XML. The default extension is “.dd”.

**/Dialogs**

This switch enables error message dialogs, which is useful for diagnostic purposes. Warnings and informational messages are written to the log only, and are not displayed in dialogs. Dialogs are disabled by default, as they may interfere with production.

**/Delete**

This switch enables the deletion of the original XML file after it has been processed. This switch is usually desirable, as it prevents old XML files from building up in the scanned directory.

**/?**

**/Help**

This switch brings up a command line help dialog for XML2TXT.

## ***Command Line Examples***

### ***Explicit XML File Name***

The following command line tells XML2TXT to process the file “Oracle.xml”, and create Commander script from it. BarTender will look in the folder “C:\OracleFormats” in order to find the correct label format file to print.

```
XML2TXT /XML="C:\Oracle.xml" /FormatFolder="C:\OracleFormats"
```

### ***Implied XML File Name***

The following command line tells XML2TXT to process whatever trigger file Commander is responding to. As with the previous command line, BarTender will look in the folder “C:\OracleFormats” in order to find the correct label format file to print.

```
XML2TXT /XML="<Trigger File Name>" /FormatFolder="C:\OracleFormats"
```

### ***Command Line Example: Diagnostic Arguments***

The following command line tells XML2TXT to write all information, warning and error messages to a log file at the root of the C: drive, and to display any error messages to the user.

```
XML2TXT /XML="<Trigger File Name>" /Log="C:\Log.txt" /Dialogs
```

(Note that the /FormatFolder argument was left out of the above command line. If you actually do this, BarTender will automatically try to find the desired label format location according to a predefined search pattern of folders to search in.)

## Appendix D: Configuring XML2TXT

(This advanced information is for users of the RFID Enterprise and base Enterprise editions. The XML2TXT utility is not needed by the Enterprise Print Server edition.)

This appendix explains which XML2TXT program settings can be customized in the file “XML2TXT.config”. This file is located in the Commander application folder and may be edited using a text editor such as Notepad or an XML editor.

### **Format Map**

If Oracle specifies a “seeded” Oracle format name in the XML label request file, XML2TXT will use a “format map” defined in the “XML2TXT.config” file to determine which label format to use. Format types are defined under the XML element “FormatMap”, and may be added, modified or deleted from this map. An entry in this map may look like this:

```
<Format Name="seeded label default for serial" File="Serial.btw" />
```

This line means that if Oracle generates an XML file referring to the “seeded label default for serial”, then XML2TXT will specify the “Serial.btw” label format name when it converts the XML into Commander script and BarTender will print using that label format.

### **Custom Commander Script Arguments**

When XML2TXT converts an XML file into Commander script, it produces the simplest Commander script possible from the XML. Depending on the information included in the XML file, XML2TXT may automatically include any or all of the following Commander script arguments:

```
/AF /PRN /PrintJobName /D /C /DbTextHeader /R
```

Additional Commander script arguments are specified in the XML2TXT configuration file under the <CommanderScriptArgs> element. These additional arguments will be added to the final Commander script.

```
<CommanderScriptArgs>/P /DD</CommanderScriptArgs>
```

The standard additional arguments are /P and /DD. These arguments mean that the label will print out normally and then BarTender will delete the Commander script file when it is done printing. Multiple arguments need to be separated by spaces.

## Appendix E: Troubleshooting XML2TXT

(This advanced information is for users of the RFID Enterprise and base Enterprise editions. The XML2TXT utility is not needed by the Enterprise Print Server edition.)

### **Log File**

You can enable message logging in XML2TXT by including the following command line argument:

```
/Log=" [Message Log Filename] "
```

Messages are written to the log file with the times at which they occur and are categorized into informational messages, warnings and errors.

### **Error Dialogs**

You can enable error message dialog boxes with the following command line switch:

```
/Dialogs
```

This feature is useful for immediately seeing any problems that might occur when running XML2TXT. Dialogs also requires fewer permissions than writing to the log file, so error dialogs may work in some cases when the log file does not. Because error message dialogs require a user's response, it's usually not desirable to leave this switch enabled during normal production operation. Only error messages are displayed in dialogs; warning and informational messages are not displayed.

### **Running XML2TXT from a Network Share**

XML2TXT is a .NET application, which means that some configuration may be necessary in order to run the executable from a network location. To run XML2TXT from a network share, follow these steps:

1. On the client computer, go to Start | Settings | Control Panel.
2. Go to Administrative Tools, and run "Microsoft .NET Framework Wizards".
3. Click the option named "Trust an Assembly".
4. Select "Make changes to this computer", then click Next.
5. Click Browse, and browse to the location of XML2TXT, or enter the path in the area provided. Click Next.
6. Move the slider all the way up to the "Full Trust" position. Click Next again.
7. Click Finish.

## **Appendix F: For More Information**

More information on Oracle WMS and MSCA is available at Oracle's website:

<http://www.oracle.com>

[http://www.oracle.com/applications/order\\_mgmt/Warehouse.html](http://www.oracle.com/applications/order_mgmt/Warehouse.html)

[http://www.oracle.com/applications/order\\_mgmt/MSCA.html](http://www.oracle.com/applications/order_mgmt/MSCA.html)

More information on the Commander utility is available in the Commander white papers:

<http://www.seagullscientific.com/aspx/whitepapers.aspx>



## Available Seagull White Papers

### General White Papers

- The Advantage of Drivers by Seagull
- Choosing the Right BarTender Edition
- Label System Security

### Companion Applications

- Printer Maestro, Part 1: Enterprise Print Management
- BarTender Security Center
- BarTender Web Print Server

### Recent Upgrades

- What's New in the Latest BarTender

### Integration White Papers

- Integration Overview
- Commander
- Commander Examples
- BarTender's .NET SDKs
- BarTender's ActiveX Automation Interface
- Exporting Printer Code Templates
- Using BarTender with Terminal Services and Citrix MetaFrame
- XML Integration with Oracle's WMS and MSCA

### Integration With SAP

- SAP Integration Methods
- Reading SAP IDocs
- SAP Auto Infrastructure Integration with BarTender

### Miscellaneous White Papers

- BarTender Enterprise Licensing
- Printing International Characters Using BarTender
- BarTender Software Activation
- Using BarTender's Application Identifier Wizard
- Optimizing Label Printing Performance
- Status Monitor Overview
- Silent Install

For downloadable versions, visit:

[www.seagullscientific.com/asp/whitepapers.aspx](http://www.seagullscientific.com/asp/whitepapers.aspx)

