

White Paper

## Commander

Utility for Cross-Platform  
and Enterprise Integration

**WHAT IS COMMANDER? ..... 3**  
    HOW COMMANDER WORKS ..... 3  
    TASKS ..... 3  
    TASK LISTS ..... 4  
    HOW COMMANDER DETECTS TRIGGERS ..... 4  
    USING COMMANDER TO RUN ANY APPLICATION ..... 4

**TAKING ADVANTAGE OF COMMANDER'S FLEXIBILITY ..... 5**  
    OVERVIEW ..... 5  
    MULTIPLE BARTENDER PROCESSES ..... 5  
    COMMANDER SCRIPTS ..... 6  
    COMMANDER VARIABLES ..... 7

## What is Commander?

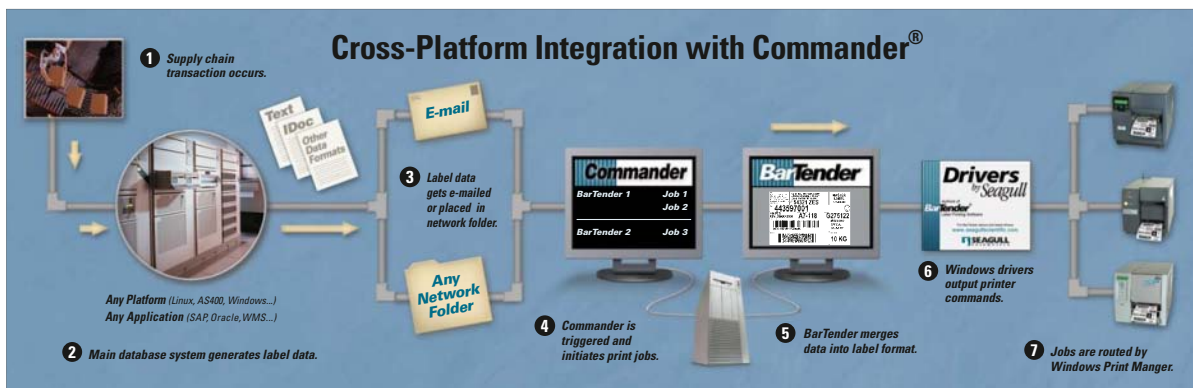
Commander is a software utility, available with the Enterprise Edition of BarTender, that enables you to perform automatic label-printing using BarTender in situations where using command line or ActiveX automation is either not possible or not cost-effective. Commander can be run as an application or, on Windows NT, 2000, XP, and 2003, as a Windows service.

### How Commander Works

When an application needs labels, it simply creates a triggering event (called a “trigger”), such as placing a file in a location of your choosing on the network or sending an e-mail to an account of your choosing. Commander detects the arrival of this file (or e-mail) and then “wakes up” BarTender so it can pour your data into the label design and automatically print your labels.

A trigger file or message may be empty; but it can also contain data that will be read by BarTender, by another application that Commander launches, or by Commander itself. Because trigger-creating applications can include different content in different triggers, the tasks Commander will execute, and the data it uses, can be defined dynamically by the application.

For example, an in-house order fulfillment application enters data about an order into a database and saves an empty file named NewOrder.dat in a directory being watched by Commander. Commander finds the file and launches BarTender with a label format that has been configured to query the database about orders entered after a specified time. BarTender reads the data and prints the label. Commander then deletes the NewOrder.dat file and resumes monitoring the directory.



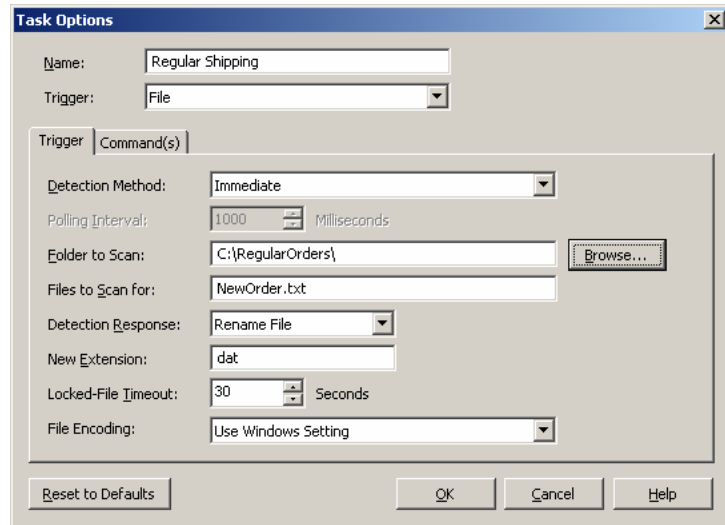
### Tasks

A Commander *task* is a set of one or more commands associated with a trigger.

There are two major parts to defining a task in Commander:

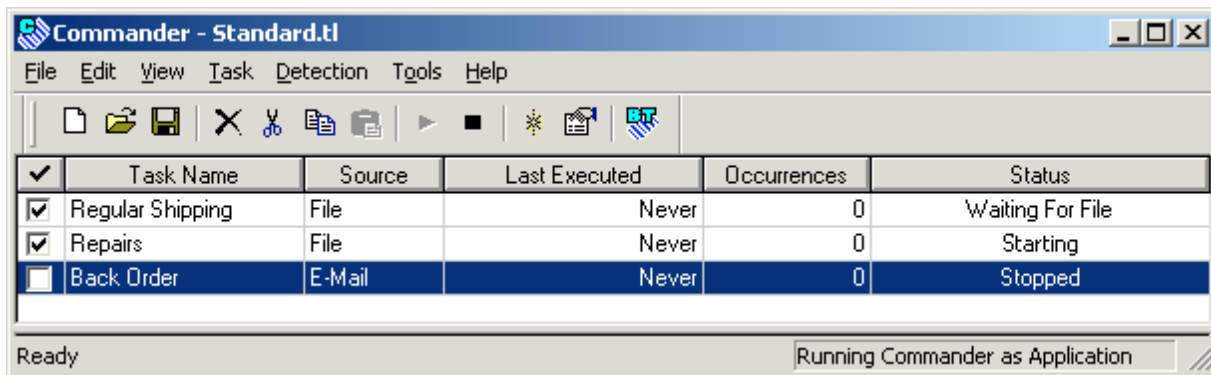
- First you select what type of trigger will cause Commander to execute the command(s) and define task options.

- Then you identify the commands that will be executed in response to the trigger. These can be BarTender commands, operating system command lines, or a Commander script. You can mix all three types of commands in a given task.



### Task Lists

Commander can monitor multiple triggers and execute multiple tasks in response to each. The set of tasks that Commander is ready to execute at any given time is called a *task list*. You may define more than one task list, and load/unload task lists as needed.



### How Commander Detects Triggers

Commander's methods of detecting triggers are customizable. For example, if a file trigger is used, you can configure Commander to detect it in either of two ways. First, have Commander detect immediately that the trigger file has been created. Second, have Commander check the target directory at periodic intervals for the presence of the trigger file. You can set the interval with precision measured in milliseconds.

### Using Commander to Run Any Application

Because Commander can execute any command that can be used with the **Run** option of the **Start** menu, it can be used to launch automatically any application on the system in response to a trigger.

For example, many businesses have online forms on their web sites. The form processing software can be configured to save each form under the name <e-mail\_address>.txt. Commander can then be set to watch for such files and, when it finds one, open an e-mail application and send an acknowledgement to the e-mail address.

## **Taking Advantage of Commander's Flexibility**

### *Overview*

Here are some of the ways that Commander can be the solution in more complex scenarios:

- Wildcards \*, and ?, can be used to specify the trigger file name (or the Subject line when an e-mail trigger is used).
- Special Commander variables can be used to pass the data to BarTender.
- Commander can be preset to use any of BarTender's command line parameters when it launches BarTender.
- The trigger file can itself contain the command line parameters that BarTender is to use when launched.
- A trigger file (or e-mail) can itself contain the data that should appear on a label.
- The trigger-creating application need not be a Windows application.
- Commander can perform more than one task in response to a given trigger.
- Commander can instruct BarTender to close after it has printed the label.
- Commander can be set to log all of its activities.

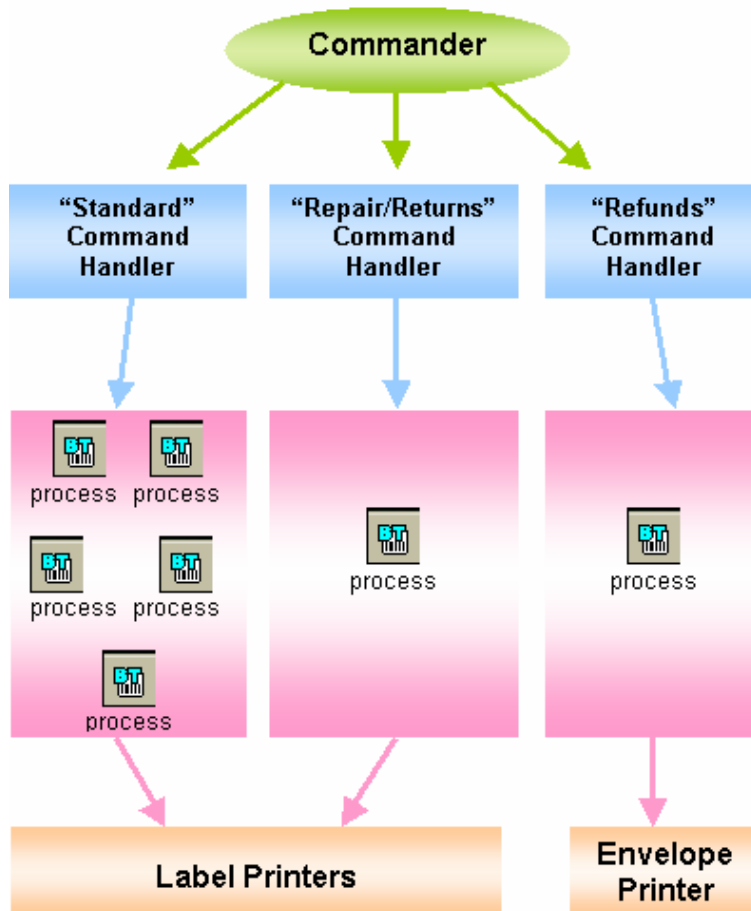
### *Multiple BarTender Processes*

You can increase the speed at which labels are printed by having multiple instances (also called processes) of BarTender running, each working simultaneously on different tasks. Moreover, different instances of BarTender can launch with different sets of command parameters so you can assign each task to a BarTender instance whose command parameters are most appropriate to the task. You can also assign tasks that are seldom triggered to launch an instance of BarTender and then close it immediately when the task has completed.

Each command in a task can specify a command handler to execute it. Command handlers are primarily responsible for preloading label formats and passing print job instructions to BarTender processes. If a particular command handler is heavily used, you can grant it more than one BarTender process.

For example, a user could create these three command handlers:

- One, called “Standard,” uses five instances of BarTender, each of which launches with command parameters that cause it to open a label format used on all new shipments and login to an ODBC database. Each time a new order is placed, a trigger is created, and Commander uses the first available BarTender instance to print the label.



- A second, called “Repair/Returns” uses just one instance of BarTender whose command line parameters cause BarTender to open a label format used on packages that contain repaired items being returned to their owners.
- Finally, a third command handler, called "Refunds" is defined to launch an instance of BarTender, print an envelope, and then close that instance immediately. This is assigned to a task that is triggered only a few times a week.

### *Commander Scripts*

A Commander script is a set of written instructions that Commander can read and execute. Applications that create triggers can add scripts to the triggers. This allows the application to give Commander instructions that vary from trigger-to-trigger, instead of having it perform exactly the same actions in response to every trigger of a given type.

The primary purpose of a Commander script is to specify the BarTender command line and to specify the data that is to populate the fields on the label.

## Commander Variables

Commander enables the use of variables on the BarTender command line, operating system command line, and in Commander scripts. The variables are replaced at task execution time.

Among other uses, these variables can be used to specify such things as label format name, printer name, text file name, IDoc file name, ODBC user name or password, and additional command parameters.

The variables are:

Variable	Definition
<b>&lt;Trigger Contents&gt;</b>	Replaced by the contents of the trigger.
<b>&lt;Trigger File Name&gt;</b>	Replaced by the complete path and filename of the trigger file.
<b>&lt;Trigger Base File Name&gt;</b>	Replaced by the trigger filename, not including either the path or the extension.
<b>&lt;Trigger File Path&gt;</b>	Replaced by the trigger file's path. Does not include the filename or extension.
<b>&lt;Time&gt;</b>	Replaced by the time the task was triggered with the format [h]h:mm:ss xM.
<b>&lt;Date&gt;</b>	Replaced by the date the task was triggered with the format [m]m/[d]d/yyyy.
<b>&lt;E-mail Subject&gt;</b>	Replaced with the entire Subject line of the e-mail message when an e-mail trigger is being used.

## Available Seagull White Papers

### General White Papers

- The Advantage of Drivers by Seagull
- Choosing the Right BarTender Edition
- What's New in the Latest BarTender

### Integration White Papers

- Integration Overview
- Getting Started with ActiveX Automation Using C#
- Getting Started with ActiveX Automation Using VB.NET
- Getting Started with ActiveX Automation Using VB6
- Commander
- Commander Examples
- Exporting Printer Code Templates
- Using BarTender with Terminal Services and Citrix MetaFrame

### SAP Integration White Papers

- SAP Integration Methods
- Reading SAP IDocs

### Miscellaneous White Papers

- BarTender Enterprise Licensing
- Printing Foreign Text Using BarTender
- Encoding RFID Tags
- BarTender Software Activation
- Using BarTender's Application Identifier Wizard
- Optimizing Label Printing Performance

For downloadable versions, visit:

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

