

Commander

Utility for Cross-Platform and Enterprise Integration



Seagull Scientific, Inc
1616 148th Ave. S.E.
Bellevue, WA 98007-6848
USA

Europe
C/ Velázquez, 15-6º-Izq.
28001 Madrid
Spain

TEL: 425-641-1408
FAX: 425-641-1599
Sales@SeagullScientific.com
www.SeagullScientific.com

TEL: +34 91 435 2525
FAX: +34 91 578 0506
eurosales@SeagullScientific.com

All screen images used in this document are for illustrative purposes only, and are intended to provide an example of the screen only. Screens may vary depending on the edition of the product being used.

Seagull Scientific, Inc. has made every effort to ensure the accuracy of information contained within this document. However, Seagull Scientific, Inc. makes no warranties with respect to this document and disclaims any implied warranties of merchantability or fitness for a particular purpose. The information in this document is subject to change without notice. Seagull Scientific, Inc. assumes no responsibility for errors that may appear in this document. All persons named in this document are fictitious. No connection between anyone living or dead is implied or intended.

All trademarks acknowledged.

In North America:

Seagull Scientific, Inc
1616 148th Ave. S.E.
Bellevue, WA 98007-6848
USA

In Europe:

C/ Velázquez, 15-6º -Izq.
28001 Madrid
Spain

August, 2003

All rights reserved. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the written permission of Seagull Scientific, Inc. Information in this document is subject to change without notice and does not represent a commitment on the part of Seagull Scientific, Inc.

BarTender is a registered trademark of Seagull Scientific, Inc.

Version: 2003-08-11 10:39

Contents

WHAT IS COMMANDER?	4
OVERVIEW	4
TASKS	5
TASK LISTS	6
HOW COMMANDER DETECTS TRIGGERS	7
USING COMMANDER TO RUN ANY APPLICATION	7
TAKING ADVANTAGE OF COMMANDER'S FLEXIBILITY	9
OVERVIEW	9
MULTIPLE BARTENDER PROCESSES	9
COMMANDER SCRIPTS	11
COMMANDER VARIABLES	12

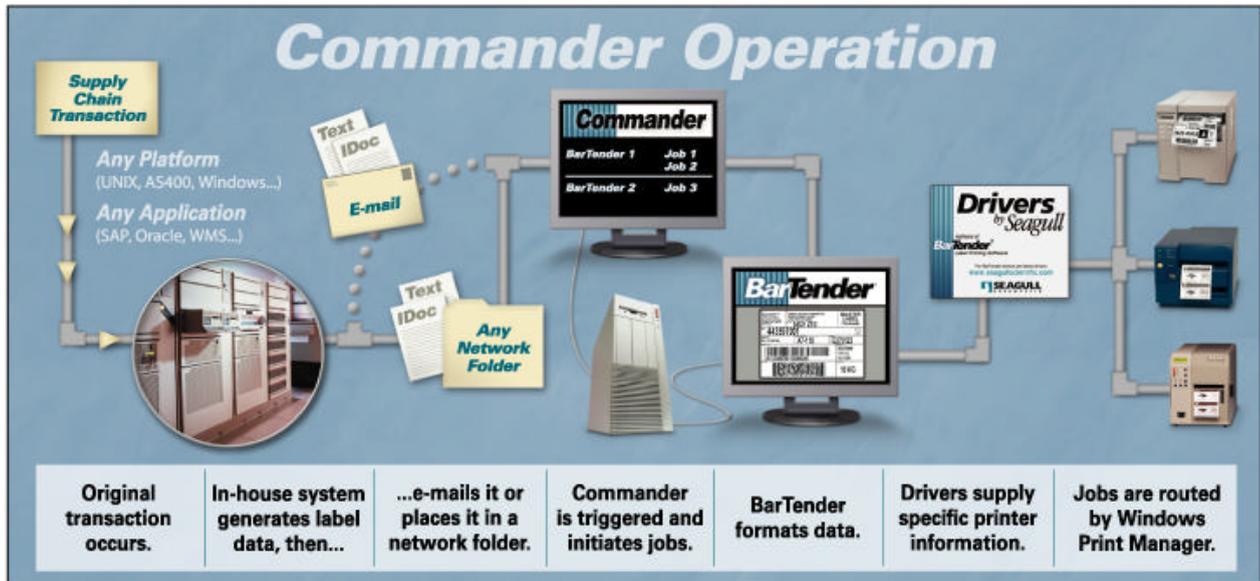
What is Commander?

Overview

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 flow 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.

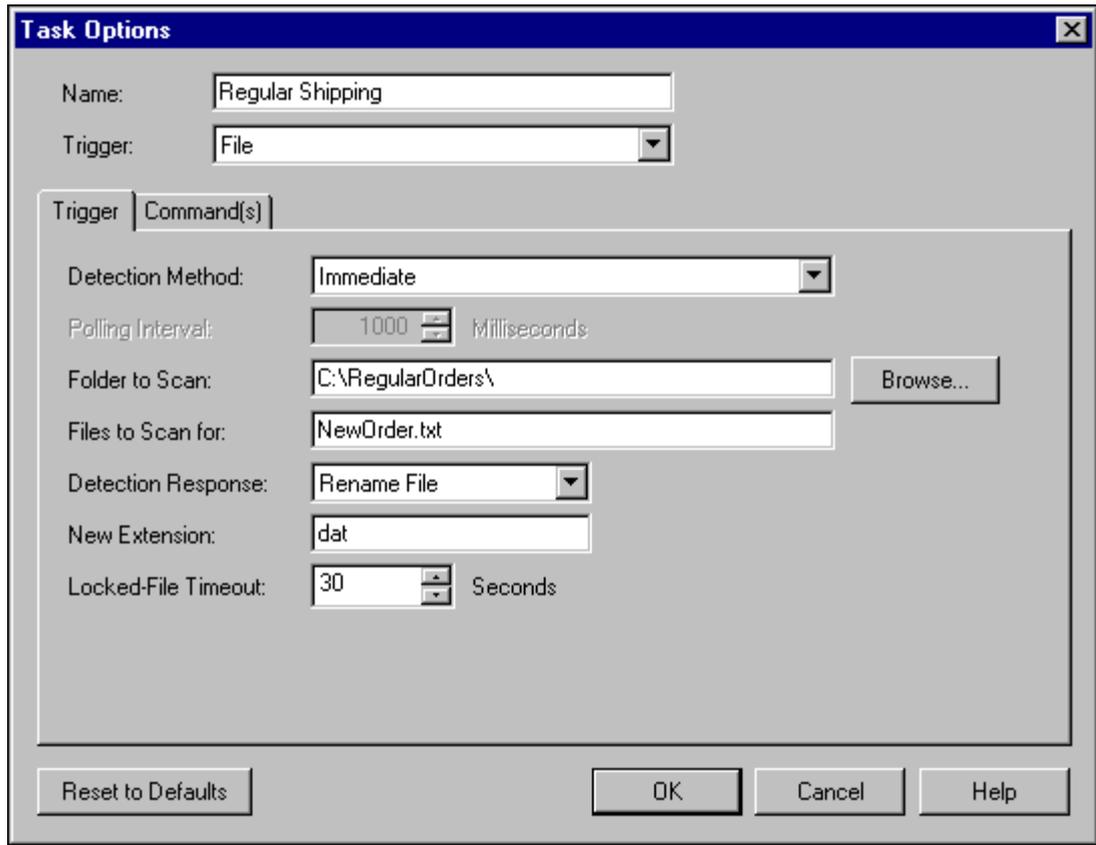


Tasks

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 Options

Name: Regular Shipping

Trigger: File

Trigger | Command(s)

Detection Method: Immediate

Polling Interval: 1000 Milliseconds

Folder to Scan: C:\RegularOrders\ Browse...

Files to Scan for: NewOrder.txt

Detection Response: Rename File

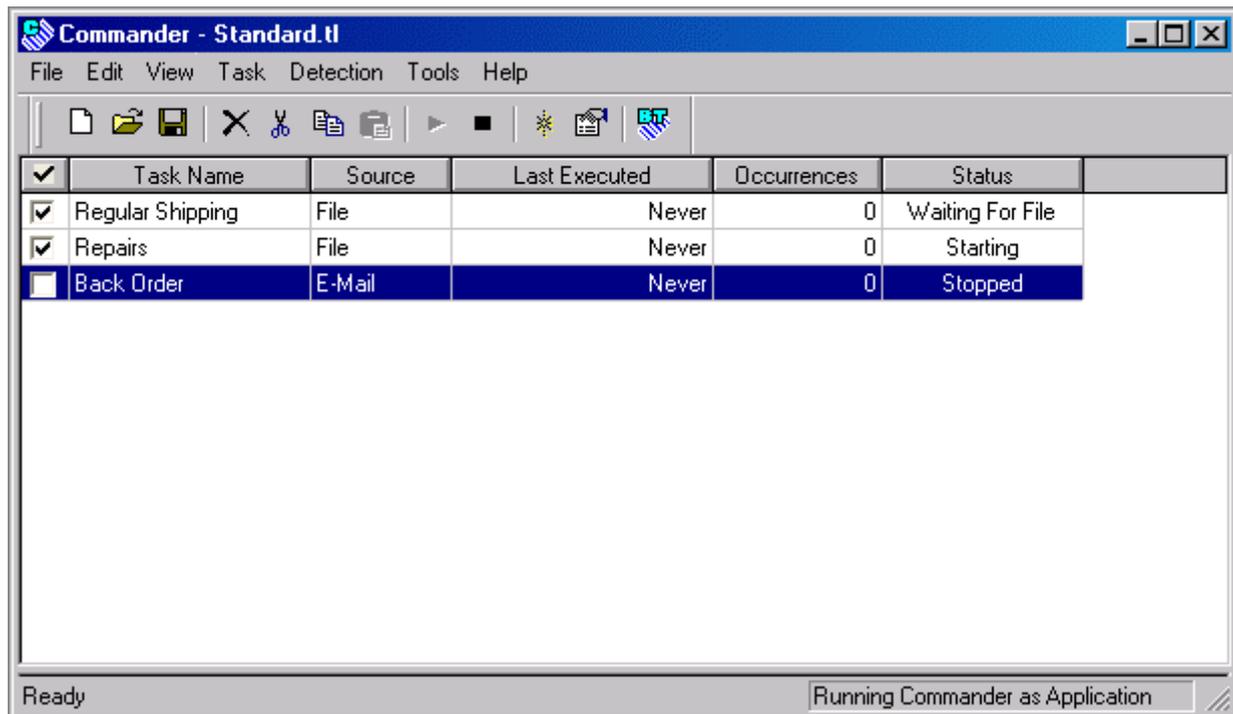
New Extension: dat

Locked-File Timeout: 30 Seconds

Reset to Defaults OK Cancel Help

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 automatically launch 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.

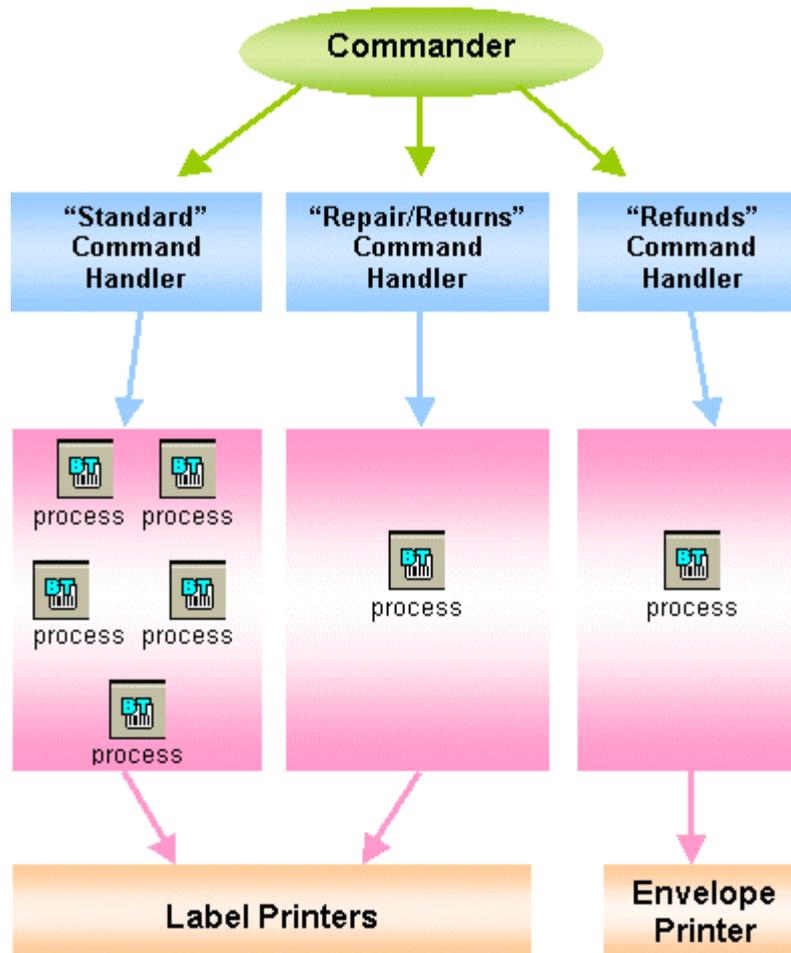
Multiple BarTender Processes

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