

***DocumentBurst* User Guide**

***DocumentBurster* User Guide**

Copyright © 2006-2013 *SourceKraft Systems & Consulting Ltd.* All rights reserved.

DocumentBurster trademark is property of *SourceKraft Systems & Consulting Ltd.*

All other marks and trademarks are properties of their respective owners.

Table of Contents

| | |
|--|----|
| Manual Conventions | vi |
| I. Introduction | 1 |
| Overview | 2 |
| <i>DocumentBurst</i> | 2 |
| <i>DocumentBurst</i> Server | 2 |
| II. <i>DocumentBurst</i> | 5 |
| 1. Burst and Merge PDF Reports | 6 |
| Burst PDF Reports | 6 |
| Steps to Follow When Bursting a Report | 7 |
| Burst a Sample Report - Invoices-Oct.pdf | 7 |
| Merge PDF Reports | 11 |
| Configuration | 13 |
| Advanced Settings | 14 |
| 2. Burst Excel Reports | 17 |
| Overview | 17 |
| Burst by Distinct Sheets | 17 |
| Burst by Distinct Column Values | 19 |
| Steps to Follow When Bursting an Excel Report by Distinct Column Values..... | 25 |
| Sample - Customers-Distinct-Column-Values.xls | 25 |
| 3. Distribute Reports | 28 |
| Overview | 28 |
| Enable Report Distribution | 28 |
| Distribute Reports by Email | 29 |
| Email Connection Settings | 29 |
| Dynamic Email Connection Settings | 30 |
| Email Each Report to One Recipient | 31 |
| Email Each Report to Multiple Recipients | 32 |
| Text Email Messages | 33 |
| HTML Email Messages | 35 |
| Attachments | 37 |
| Configure Attachments | 37 |
| Upload Reports | 39 |
| <i>cURL</i> Integration | 39 |
| FTP | 40 |
| File Share | 41 |
| FTPS | 42 |
| SFTP/SSH/SCP | 43 |
| HTTP/WebDAV | 45 |
| 4. Configuration Templates | 47 |
| Process PDF Reports Using a Custom Configuration File | 47 |
| Process Excel Reports Using a Custom Configuration File | 48 |
| Add, Edit and Delete Configuration Templates | 48 |
| Create Similar Configuration Templates | 49 |
| 5. Variables | 51 |
| Built-In Variables | 52 |
| Sample - Payslips.pdf | 53 |
| User-Defined Variables | 54 |
| Sample Invoices-Oct.pdf - Customizable Burst File Name | 55 |
| Excel User-Defined Variables | 58 |
| 6. Automatic Polling for Incoming Reports | 60 |
| Watch a Folder for Incoming Reports | 60 |

| | |
|--|-----|
| 7. Quality Assurance | 62 |
| Test for Correctness | 62 |
| Configuration | 62 |
| Run Test | 63 |
| Logging | 64 |
| Send Processing Errors by Email | 66 |
| Quarantine Reports Which Fail To Be Delivered | 66 |
| Error Handling | 67 |
| 8. Command Line | 69 |
| Usage | 69 |
| Burst Reports | 69 |
| Merge Reports | 70 |
| Poll Folder for Incoming Reports | 70 |
| Quality Assurance | 71 |
| III. DocumentBuster Server | 73 |
| 9. DocumentBuster Server | 74 |
| Installation | 74 |
| Prerequisites | 74 |
| Download <i>DocumentBuster Server</i> | 74 |
| Start and Stop | 75 |
| A. Server | 75 |
| B. Web Console | 75 |
| Web Console | 76 |
| Screenshots | 76 |
| Run <i>DocumentBuster</i> at System Startup | 79 |
| Windows Services | 79 |
| Windows Scheduling | 82 |
| Scheduling | 84 |
| Configuration | 85 |
| IV. Appendices | 86 |
| A. How to Do This and That | 87 |
| How To: Distribute External Reports? | 87 |
| Configuration | 87 |
| Recipients List and Document(s) Paths | 88 |
| How To: Implement Conditional Report Distribution? | 90 |
| skip_current_file_distribution_if.groovy | 91 |
| B. SharePoint Report Delivery | 92 |
| Distribute Reports to SharePoint by Email | 92 |
| SharePoint Configuration | 92 |
| Sample - Burst samples/Payslips.pdf Report to SharePoint 2007 | 93 |
| C. <i>DocumentBuster</i> Performance | 102 |
| Performance Guidelines - Getting the Most From Your <i>DocumentBuster</i> System | 102 |
| Performance Monitoring | 103 |
| Enable Performance Monitoring | 103 |
| Built-in <i>DocumentBuster</i> Performance Monitoring | 104 |
| Integration with Professional Third Party Monitoring Tools | 107 |
| Further Reading | 108 |
| D. Troubleshooting | 109 |
| Overview | 109 |
| Quick & Professional Support | 109 |
| Common Problems | 109 |
| I Get Only 25 Reports | 109 |
| Issues Running Basic Features? | 109 |
| UnsupportedClassVersionError Exception? | 109 |

| | |
|--|-----|
| Mess With Multiple Java Installations | 110 |
| Windows - DocumentBuster.exe GUI is Failing to Start? | 110 |
| Windows - DocumentBuster.exe GUI Still Fails? | 110 |
| Windows - DocumentBuster.exe GUI Still Fails? | 110 |
| Windows - DocumentBuster.exe GUI Still Fails? | 110 |
| Windows - DocumentBuster.exe GUI Comes Up For a Second and Then Disappears? | 111 |
| Burst Issue 1 | 111 |
| Burst Issue 2 | 111 |
| Variable Values Are Not Parsed Correctly? | 111 |
| Windows - DocumentBuster Server Is Failing to Start? | 112 |
| Windows - DocumentBuster Server Is Not Processing the Web Console Jobs? | 112 |
| Windows - DocumentBuster Web Console Is Failing to Start? | 112 |
| Windows Service Stops on Windows Server 2003? | 113 |
| Email Is Failing? | 114 |
| Email Still Fails? | 114 |
| Email SMTP Error 421 | 114 |
| Email SMTP Error 550 5.7.1 Unable to relay | 115 |
| Few Emails Are Going and Then Email Stops Working | 116 |
| Email Is Still Failing? | 116 |
| Upload Issue? | 117 |
| I messed up with the configuration and now the program is failing. Is it possible to restore the default configuration? | 117 |

Manual Conventions

1. Path Separator

This document uses slash character (/) to display directory and file components of a path.

Microsoft Windows can accept either the backslash (\) or slash (/) characters to separate directory and file components of a path, while the Microsoft convention is to use a backslash (\). Since *DocumentBurst* is intended to work on multiple operating systems (e.g. Windows, Linux), the convention for this manual is to use the slash character (/) to display the components of a path.

Part I. Introduction

Welcome to *DocumentBurst* User Guide

This document will help the reader to get productive with *DocumentBurst* in order to burst and distribute business reports. We'll also offer tips on sending personalized e-mails, rich formatted HTML e-mails and on getting the maximum performance from your *DocumentBurst* report delivery system.

As for the system requirements, in most of the situations, *DocumentBurst* will perform well on any low-end machine which can decently run Windows XP. However, if your enterprise organization is expected to process really huge reports then a more powerful machine will help.

What to Expect

In this user guide, you'll learn how to

- Burst and merge PDF reports
- Burst Microsoft Excel reports
- Distribute reports (e.g. payslips, invoices, etc.) through personalized email messages or through FTP
- Execute and automate *DocumentBurst* through command line
- Windows Services - install and run *DocumentBurst Server* as standard Windows Services in order to achieve unattended document delivery
- Schedule reports to be processed and distributed at a specific date and time, to distribute them hourly, daily, weekly, monthly or set up your own frequencies, every other day, third Monday of the month and so on

Quick & Professional Support

If you have any questions that aren't answered here, feel free to contact us: support@pdfburst.com

Feedback

We welcome feedback on our products, including this manual. If you would like to make any suggestions for improving our products, please contact us: support@pdfburst.com

Now let's get started.

Overview

DocumentBurst is a powerful solution to schedule, break up (split), merge and distribute reports and can naturally complement any current business intelligence deployment by adding advanced report delivery capabilities.

DocumentBurst can process reports generated with an existing in-house reporting platform or with any of the well known leading report writers including Crystal Reports and Web Intelligence (Business Objects/SAP), IBM Cognos, Oracle Hyperion, Microsoft Reporting Services, Microsoft Access, QlikView, JasperReports, Eclipse BIRT, Pentaho, etc.

DocumentBurst software is currently used successfully together with business applications like Oracle PeopleSoft, SAP ERP, Oracle Applications, Sage Applications, Microsoft Dynamics, JD Edwards, MYOB and QuickBooks accounting software.

The software has two flavors

- ***DocumentBurst*** has very good report bursting and report distribution capabilities. *DocumentBurst* has a desktop based GUI interface which can be used by a single user.
- ***DocumentBurst Server*** has all the capabilities of *DocumentBurst* and some additional and advanced features like scheduling, capability to run as windows services and a browser based web interface which can be accessed simultaneously by multiple people.

DocumentBurst

DocumentBurst can process PDF or Excel reports.

- The software can break up reports such as pay slips, invoices or statements for each of your employees, customers or partners.
- *DocumentBurst* can distribute reports to a wide range of destination types such as Email, FTP, FTPS, SFTP, TFTP, Windows shared drives, Unix Samba servers, WebDAV servers and document management systems.
- The software can publish reports to enterprise portals such as Microsoft Sharepoint Server, SAP NetWeaver, Oracle Portal or IBM WebSphere Portal.
- *DocumentBurst* can generate custom dynamic email messages based on email templates.
- The software can generate good looking HTML email messages based on HTML email templates.

DocumentBurst Server

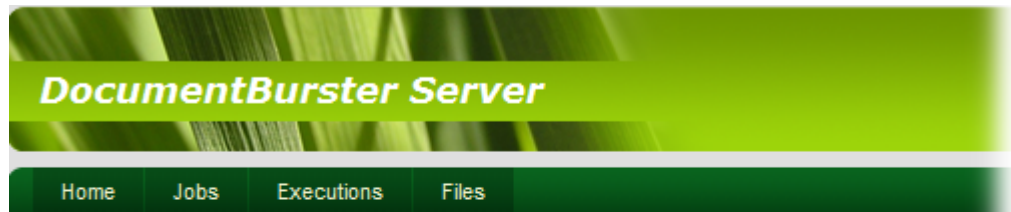
DocumentBurst Server has all the features which *DocumentBurst* has, together with some additional advanced capabilities like scheduled report distribution and unattended report delivery in order to achieve the most complex report delivery scenarios.

DocumentBurst Server can be executed like a long-running Windows service in order to perform report distribution with minimum or no user intervention. *DocumentBurst Server* services can be configured to start when Windows is booted (and run in the background as long as Windows is running), or they can be started manually, when required.

DocumentBurster Server is a fully fledged report distribution solution which can be tailored with advanced features to perfectly meet the most complex report bursting and report distribution requirements.

Following capabilities are all achievable with *DocumentBurster Server*, either as out of the box features, either through the tailoring of the software

- *Distribution Server* - The software can be deployed as a central report bursting and report distribution platform which can be simultaneously accessed by multiple people or legacy applications from within your organization.
- *Cross browser web interface* - *DocumentBurster Server* has a web based interface available for IE, Firefox, Chrome, Safari and Opera.
- *Unattended report distribution* - *DocumentBurster Server* can be configured to run like standard Windows services in order to automatically burst and deliver the reports without human intervention.
- *Easy to integrate* - *DocumentBurster Server* can be easily integrated with existing CRM and ERP kind of applications.
- *Scheduling* - Using *DocumentBurster Server* it is possible to define simple or complex schedules for executing nightly, weekly or monthly report bursting and report distribution jobs.
- *Parallel report delivery* - *DocumentBurster Server* has support for parallel report distribution which allows achieving a high throughput of reports which can be distributed in a short period of time.
- *Process any report format* - Out of the box *DocumentBurster Server* can burst and distribute PDF and Excel reports. If required, the software can be tailored to merge, burst and distribute any report format in addition to the already supported PDF and Excel report types.
- *Organize, store and index the distributed reports* - *DocumentBurster Server* can be customized to support advanced storing, indexing and searching capabilities for the distributed reports.
- *Custom deployment model* - *DocumentBurster Server* can be deployed as a standalone server, or it can be deployed on various application servers such as Weblogic, Websphere, JBoss etc.



DocumentBuster Server

Home

Jobs

Executions

Files

Welcome to *DocumentBuster Server*!

Introduction

DocumentBuster is a report bursting software for processing reports such as payslips, invoices, contracts, etc.

***DocumentBuster* features**

- Simple to set up and easy to use
- Burst or break up PDF and Microsoft Excel reports
- Split the reports for each of your employees, clients and partners in order to distribute them
- Report bursting software which works with any reporting software including Crystal Reports
- Easy to integrate report bursting and report delivery capabilities into existing in-house or QuickBooks accounting software
- Report delivery for the leading open source business intelligence solutions such as Pentaho
- [Read the complete list of *DocumentBuster* features](#)

***DocumentBuster* Documentation**

- [DocumentBuster documentation](#) will help you to understand what & how can be achieved

Part II. *DocumentBurst*

The sections in Part II present how to configure *DocumentBurst* in order to burst and merge reports, distribute reports through personalized messages, automate report processing through report polling and through the command line interface or how to check the status of the processed reports through auditing and tracing.

Chapter 1, *Burst and Merge PDF Reports*

Chapter 2, *Burst Excel Reports*

Chapter 3, *Distribute Reports*

Chapter 4, *Configuration Templates*

Chapter 5, *Variables*

Chapter 6, *Automatic Polling for Incoming Reports*

Chapter 7, *Quality Assurance*

Chapter 8, *Command Line*

Chapter 1. Burst and Merge PDF Reports

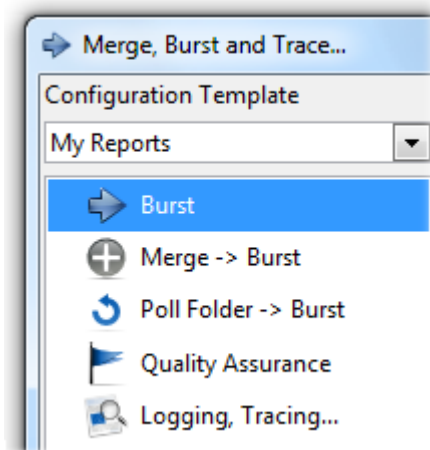
Burst PDF Reports

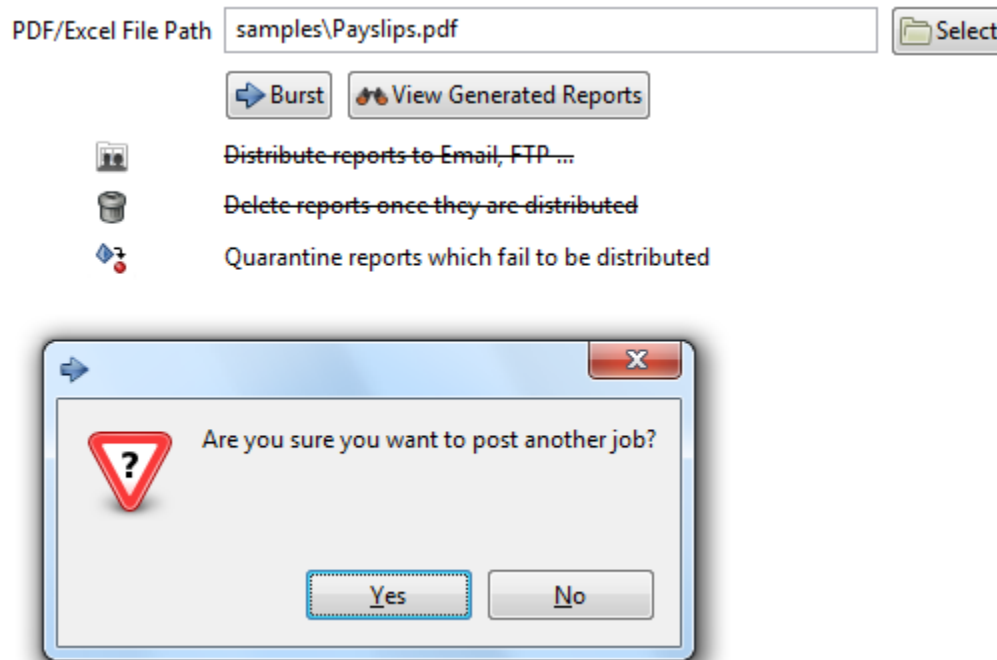
DocumentBuster is splitting the reports with the help of *burst tokens*. A *burst token* can be anything which is uniquely identifying the document to be extracted such as the invoice ID, customer number or the email address where the document should be distributed.

A burst token is simply any string that is delimited by curly brackets, “{“ and “}”. The burst tokens must be inserted into the report pages before feeding the PDF report to *DocumentBuster* for processing. *DocumentBuster* screens the document for burst tokens, and whenever it finds a new/different burst token it breaks the document.

An example of such token might be {alfreda.waldback@northridgehealth.org}. If a page from the report needs to be extracted in a separate document then *DocumentBuster* will be informed about this by using *burst tokens*. Please take a look at *Payslips.pdf* report which is available in *samples* folder. Bursting this report will generate three output files alfreda.waldback@northridgehealth.org, clyde.grew@northridgehealth.org and kyle.butford@northridgehealth.org - for the three distinct employees which are found in the input document.

In *DocumentBuster* main window select Actions , Merge, Burst and Trace... , Burst





After bursting is finished, in the output folder, following new files will be created

alfreda.waldback@northridgehealth.org
 clyde.grew@northridgehealth.org
 kyle.butford@northridgehealth.org

Steps to Follow When Bursting a Report

Following general steps should be followed when bursting a report using *DocumentBuster*.

- **Step 1** - Identify the appropriate *burst token* for the type of report which is being burst. A *burst token* can be anything which is uniquely identifying the document to be extracted such as the invoice ID, customer number or the email address where the document should be distributed.
- **Step 2** - Modify the input report to include the curly brackets, “{“ and “}”, around the previously identified burst token.
- **Step 3** - Optionally (many times the default configuration is enough), configure *DocumentBuster* to better meet your bursting needs.
- **Step 4** - Burst the report and check the output files.

Burst a Sample Report - Invoices-Oct.pdf

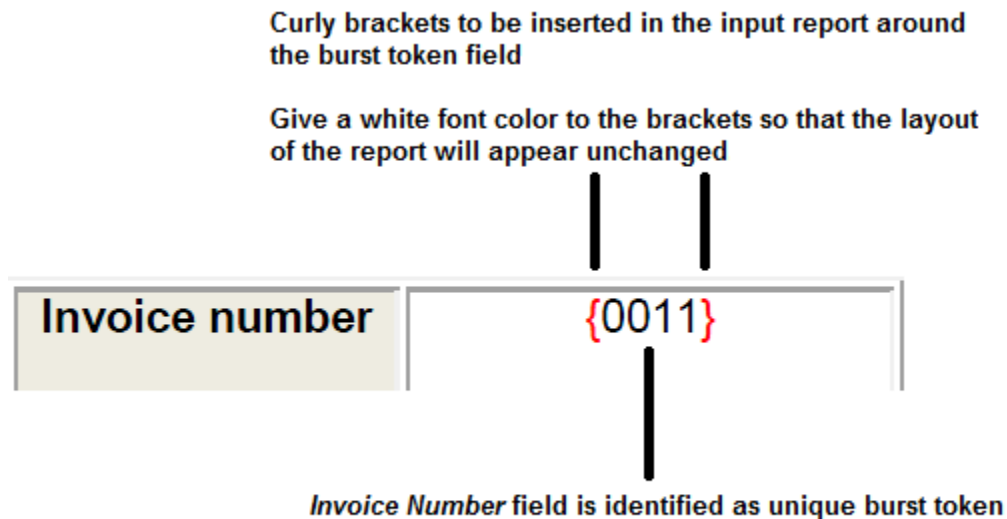
The previous steps will be exemplified when bursting the existing `samples/Invoices-Oct.pdf`.

Step 1 - Identify the appropriate *burst token*.

Since we are bursting an invoice report, the burst token is identified as being **Invoice number**. Invoice number is uniquely identifying each separate invoice.

Step 2 - Modify the input report to include the curly brackets, “{“ and “}”, around the previously identified burst token (invoice number).

The report generation software should properly fill the burst tokens into the pages of the reports. Please check `samples/Invoices-Oct.pdf` to see how it was modified to include the curly brackets, “{“ and “}”, around invoice number/burst token report field.



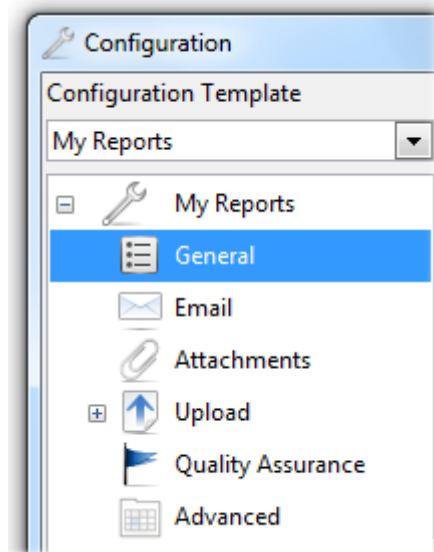
Step 3 - Configure *DocumentBurst*.




Custom Burst File Name

By default *DocumentBurst* is generating the output file names using the `$burst_token$`. `$input_document_extension$` configuration. For example, for the previous burst token/invoice number `{0011}`, the generated file name will be `0011.pdf`.

The requirement is to customize the output file names to be similar with `Invoice-0011.pdf`, for the previous burst token/invoice number `{0011}`.

In *DocumentBurst* main window select *Actions*, *Configure*, *General*



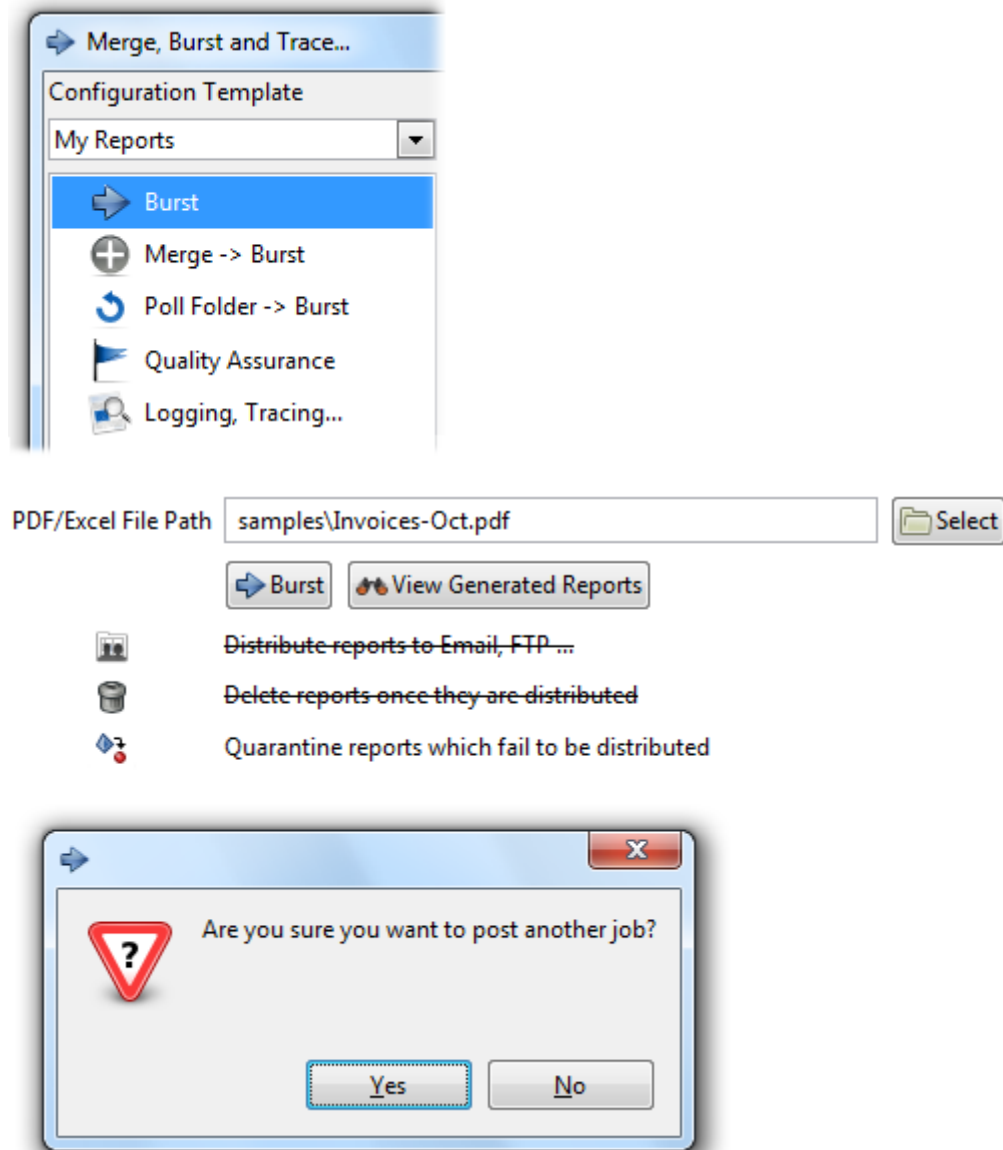
| | |
|---|---|
| Burst File Name | Invoice-\$burst_token\$.input_document_extension\$ |
| Default Merge File Name | merged.pdf |
| Output Folder | output/.input_document_name\$/.now;format="yyyy.M |
| Backup Folder | backup/.input_document_name\$/.now;format="yyyy.M |
| Quarantine Folder (should be a local folder) | quarantine/.input_document_name\$/.now;format="yy |
| Poll Folder (should be an existing local folder) | poll |
|  | <input type="checkbox"/> Distribute reports to Email, FTP ... |
|  | <input type="checkbox"/> Delete reports once they are distributed |
|  | <input checked="" type="checkbox"/> Quarantine reports which fail to be distributed |

Change the value of the *Burst File Name* configuration to be *Invoice-\$burst_token\$.input_document_extension\$*





Please see the section called “Configuration” for detailed description of *DocumentBurster* configurations.

Step 4 - Burst the report and check the output files.

In *DocumentBurster* main window select Actions , Merge, Burst and Trace... , Burst



After bursting is finished, in the output folder, following new files will be created

| Name | |
|---|--------------|
|  | Invoice-0011 |
|  | Invoice-0012 |
|  | Invoice-0013 |
|  | Invoice-0014 |

Note 1

Depending on the business requirements, the report generation software should properly fill the burst tokens into the pages of the reports.

Note 2

Use a white font color for the burst tokens (or curly brackets, “{“ and “}”) so that the visual appearance and the layout of the report will not be affected.

Note 3

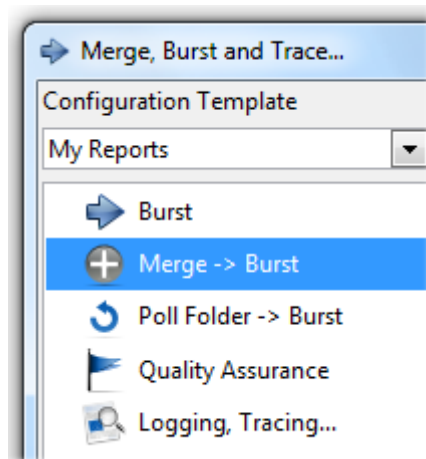
Out of the box *DocumentBurster* supports bursting of PDF and Microsoft Excel reports. If there is a need to burst other report formats such as Word reports, Open Office reports, Microsoft PowerPoint presentations (or any other document type) than *DocumentBurster* software can be tailored to achieve the bursting and distribution of such report types.

Merge PDF Reports

Sometimes, prior to bursting, it might be required to merge few reports together and burst the merged result; or you might as well just want to merge few reports or documents.

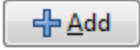




DocumentBurster can merge reports through both the command line interface and through the GUI.

In *DocumentBurster* main window select Actions , Merge, Burst and Trace... , Merge -> Burst




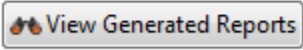
Select PDF Reports To Merge




| Name | Path |
|------------------|--------------------------|
| Invoices-Oct.pdf | samples\Invoices-Oct.pdf |
| Invoices-Nov.pdf | samples\Invoices-Nov.pdf |
| Invoices-Dec.pdf | samples\Invoices-Dec.pdf |

 Add
 Remove
 Up
 Down
 Clear

Merge -> Burst -> Distribute

Merged File Name:

☒ Burst Merged File
  Run
  View Generated Reports

 ~~Distribute reports to Email, FTP ...~~
 ~~Delete reports once they are distributed~~
 ~~Quarantine reports which fail to be distributed~~

- By default the reports are merged in the selection order. The merge order can be changed using *Up* and *Down* button commands.
- *Merged File Name* - Name of the merged file. Default value is merged .pdf
- *Burst Merged File* - If checked, the generated merged file will be also split.
- *View Generated Reports* - Allow browsing of burst/merged reports.

Following properties can be configured in *DocumentBuster* configuration screen.

- *Distribute reports to Email, FTP ...* - If *strikeout*, the software will not distribute the burst reports.
- *Delete reports once they are distributed* - If *strikeout*, the software will not delete the reports once they are distributed.
- *Quarantine reports which fail to be distributed* - If *strikeout*, the software will not quarantine the reports which fail to be distributed.

Note 1

Select multiple files - Using *Ctrl* key + *left mouse click* it is possible to select multiple files at once.

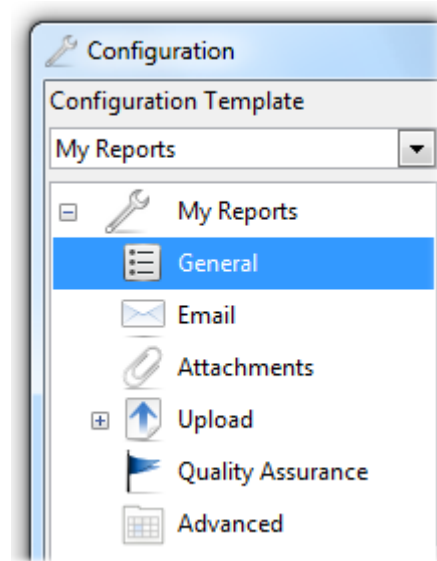
Note 2




Out of the box *DocumentBuster* can merge PDF reports. If there is a need to merge any other report format (including Word, Excel and any other document type) than *DocumentBuster* can be customized to process any report type.

Configuration

Following settings can be configured in regards with merging and bursting of the reports in *DocumentBuster*.

In *DocumentBuster* main window select Actions , Configure , General



| | |
|---|--|
| Burst File Name | <input type="text" value="\$burst_token\$.Sinput_document_extension\$"/> |
| Default Merge File Name | <input type="text" value="merged.pdf"/> |
| Output Folder | <input type="text" value="output/\$input_document_name\$/\$now;format='yyyy.MM'"/> |
| Backup Folder | <input type="text" value="backup/\$input_document_name\$/\$now;format='yyyy.MM'"/> |
| Quarantine Folder | <input type="text" value="quarantine/\$input_document_name\$/\$now;format='yyyy.MM'"/> |
| <i>(should be a local folder)</i> | |
| Poll Folder | <input type="text" value="poll"/> |
| <i>(should be an existing local folder)</i> | |
|  | <input type="checkbox"/> Distribute reports to Email, FTP ... |
|  | <input type="checkbox"/> Delete reports once they are distributed |
|  | <input checked="" type="checkbox"/> Quarantine reports which fail to be distributed |

- *Burst File Name* – Name for the generated files. Default value is *\$burst_token\$.Sinput_document_extension\$*.

For example, if the token is *{clyde.grew@northridgehealth.org}* then file name will be *clyde.grew@northridgehealth.org.pdf*.

- *Default Merge File Name* – Name for the merged file. It can be overridden for each individual merge job. Default value is `merged.pdf`.
- *Output Folder* – Used to specify the folder where to place the generated files. Default value is `output/$input_document_name$/$now; format="yyyy.MM.dd_HH.mm.ss"$`.
- *Backup Folder* – Used to specify the folder where to backup the input files. Default value is `backup/$input_document_name$/$now; format="yyyy.MM.dd_HH.mm.ss"$`.
- *Quarantine Folder* – Used to specify the folder where to quarantine the files which fail to be distributed. It should be a local folder and the default value is `quarantine/$input_document_name$/$now; format="yyyy.MM.dd_HH.mm.ss"$`.
- *Poll Folder* – Used to specify the folder to be polled for incoming reports. It should be an existing local folder and the default value is `poll`.
- *Distribute reports to Email, FTP ...* - If checked, the generated reports will be distributed as part of the bursting process, otherwise no. Default value is unchecked.
- *Delete reports once they are distributed* - If checked, the generated reports will be deleted from the disk once they are distributed, otherwise no. Default value is unchecked.
- *Quarantine reports which fail to be distributed* - If checked, the reports which fail to be distributed will be saved to the quarantine folder, otherwise no. Default value is checked.

Note

`$burst_token$`, `$input_document_extension$`, `$input_document_name$` and `$now; format="yyyy.MM.dd_HH.mm.ss"$` are variables and will be replaced at run-time with the value of the token used to burst the report, the extension of the input file, the name of the input file and the current date, formatted. For more details about variables please read Chapter 5, *Variables*

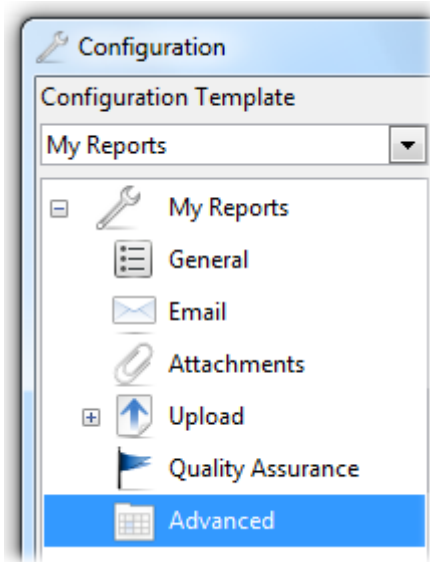
Advanced Settings

Following advanced settings can be further configured in order to allow greater control over how *DocumentBuster* works.

Note

In most of the situations the default values will get the job done without the need to change any configuration here.

In *DocumentBuster* main window select Actions , Configure , Advanced



Start Delimiter (Burst Token)

End Delimiter (Burst Token)

Number of User Variables

Number of Parallel Distribution Jobs



☐ Reuse Last Token Found in Previous Pages
(when not available in the current page)



☒ Strict Email Address Validation

- *Start Delimiter (Burst Token), End Delimiter (Burst Token)* – Start and end delimiters used to parse the burst tokens. Default values are { and } characters.
- *Number of User Variables* - The number of user variables which *DocumentBuster* will recognize and parse. Default value is 10.
- *Number of Parallel Distribution Jobs* - The number of parallel jobs which *DocumentBuster* will use when distributing the reports. Default value is 1.
- *Reuse Last Token Found in Previous Pages (when not available in the current page)* - Configuration which should be enabled when the burst token is provided only on the first (master) page of the report to be extracted and not on all the subsequent pages. Default value is *unchecked* (which means that, out of the box, *DocumentBuster* expects the burst token to be found on each and every page of the report to be extracted).
- *Strict Email Address Validation* - Configuration which, if enabled, will instruct *DocumentBuster* to validate that all the email addresses are in the well known format *string@otherstring.ext* Default value is *checked*.

Sometimes (while testing locally) it might be useful to have less strict email validation and to allow *DocumentBuster* to distribute emails to non standard (local) email addresses (e.g. *documentbuster@localhost*)

Chapter 2. Burst Excel Reports

Overview

DocumentBurster can burst and split reports which are generated for the following versions of Microsoft Excel spreadsheet application

- Excel 2010 (version 14) included in Office 2010
- Excel 2007 (version 12) included in Office 2007
- Excel 2003 (version 11) included in Office 2003
- Excel 2002 (version 10) included with Office XP
- Excel 2000 (version 9.0) included with Office 2000
- Excel 97 (version 8.0) included with Office 97

Although it does complex report processing, *DocumentBurster* doesn't require Microsoft Excel application to be installed on the machine where *DocumentBurster* is deployed. Even more, being a cross platform software, *DocumentBurster* can process Excel reports on a Windows or on any UNIX/Linux machine. A common installation scenario is to have *DocumentBurster* deployed on a separate *dedicated server machine* which is responsible for doing asynchronous batch report bursting and scheduled report distribution.

In order to achieve more business scenarios, *DocumentBurster* supports two different ways of bursting Excel reports

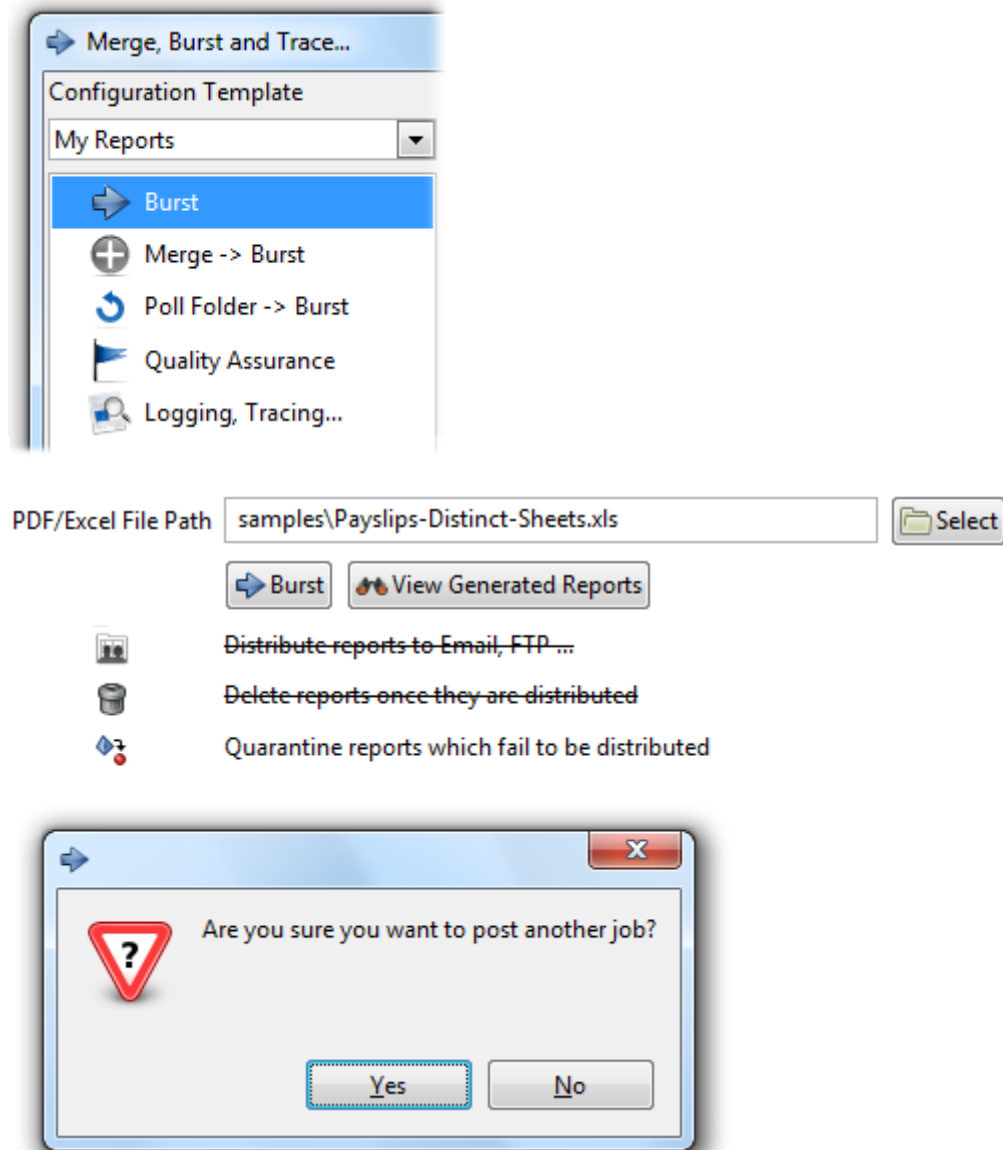
- Burst by Distinct Sheets
- Burst by Distinct Column Values

Depending on each specific business situation, one or the other of the two distinct bursting methods will be more appropriate. Following paragraphs will give details about each of the supported bursting methods.

Burst by Distinct Sheets

When bursting Excel reports using this method, *DocumentBurster* will generate a separate output file for each Excel sheet which is found in the input report.

The best way to describe this bursting method is by looking at `samples/Payslips-Distinct-Sheets.xls` sample report. This report is provided with *DocumentBurster* and contains monthly income data for three fictional employees. The data for each employee is found on a separate sheet and the name of the sheet is the employee's email address.



The input report has the following three distinct sheets *cgrew@northridgehealth.org*, *kbutford@northridgehealth.org* and *awaldback@northridgehealth.org* which, after bursting, will generate the following three distinct files



Using email addresses as sheet names for the input report can help when distributing the generated output files by email (if the respective email addresses would actually be valid and existing email addresses). For more details about distributing the generated output reports by email please check Chapter 3, *Distribute Reports*.

This bursting method is straightforward and there is not much more to discuss about it. Following paragraph will present a more complex report bursting method which can *generate a separate output file for each distinct value found in a specified column from a specified sheet*.

Note 1

By default, *DocumentBuster* will burst Excel reports by distinct sheets and the names of the sheets will be used as burst tokens.

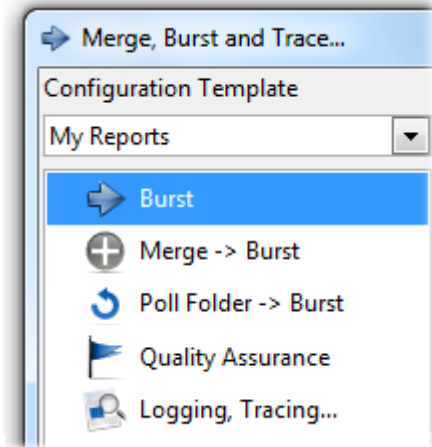
Note 2

Beside payslips, *DocumentBuster* can be configured to break up and distribute any other Excel report types such as invoices, statements, purchase orders, commissions reports or dunning letters.


Burst by Distinct Column Values


When bursting Excel reports using this method, *DocumentBuster* will generate a separate output file for each distinct value found in a specified column from a specified sheet.


Please check the existing sample report `samples/Customers-Distinct-Column-Values.xls` which can be found in `samples` folder. `Customers-Distinct-Column-Values.xls` report contains a list of customers from various countries which, after bursting, *DocumentBuster* will generate a separate file with each file containing the customers for the relevant country.

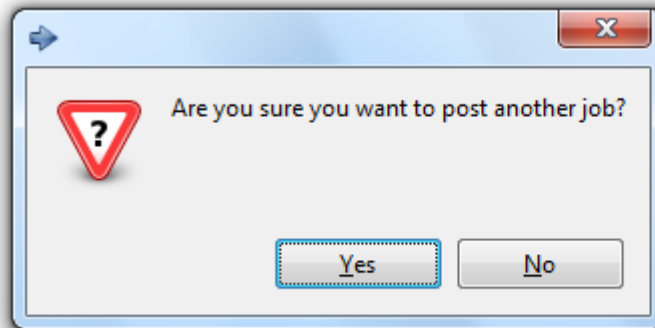


PDF/Excel File Path

 Distribute reports to Email, FTP ...

 Delete reports once they are distributed

 Quarantine reports which fail to be distributed
























Each configured *burst token* will generate a different output file.

Each *burst token* will generate a different output file.

(*)burstTokens

| |
|-------------|
| Germany |
| Mexico |
| UK |
| Sweden |
| Germany |
| France |
| Spain |
| Canada |
| Argentina |
| Switzerland |
| Brazil |
| Austria |
| Italy |
| Portugal |
| USA |
| Venezuela |
| Ireland |
| Belgium |
| Norway |
| Denmark |
| Finland |
| Poland |

| Name ▲ | Size | Type |
|---|-------|------------------------|
|  Argentina.xls | 76 KB | Microsoft Excel Wor... |
|  Austria.xls | 76 KB | Microsoft Excel Wor... |
|  Belgium.xls | 76 KB | Microsoft Excel Wor... |
|  Brazil.xls | 77 KB | Microsoft Excel Wor... |
|  Canada.xls | 76 KB | Microsoft Excel Wor... |
|  Denmark.xls | 76 KB | Microsoft Excel Wor... |
|  Finland.xls | 76 KB | Microsoft Excel Wor... |
|  France.xls | 77 KB | Microsoft Excel Wor... |
|  Germany.xls | 77 KB | Microsoft Excel Wor... |
|  Ireland.xls | 76 KB | Microsoft Excel Wor... |
|  Italy.xls | 76 KB | Microsoft Excel Wor... |
|  Mexico.xls | 77 KB | Microsoft Excel Wor... |
|  Norway.xls | 76 KB | Microsoft Excel Wor... |
|  Poland.xls | 76 KB | Microsoft Excel Wor... |
|  Portugal.xls | 76 KB | Microsoft Excel Wor... |
|  Spain.xls | 77 KB | Microsoft Excel Wor... |
|  Sweden.xls | 76 KB | Microsoft Excel Wor... |
|  Switzerland.xls | 76 KB | Microsoft Excel Wor... |
|  UK.xls | 77 KB | Microsoft Excel Wor... |
|  USA.xls | 77 KB | Microsoft Excel Wor... |
|  Venezuela.xls | 76 KB | Microsoft Excel Wor... |

How It Works

The *last sheet* of the input report `samples/Customers-Distinct-Column-Values.xls` is called ***burst*** and contains the metadata required for bursting the report.

Note 1 - *burst* Sheet

In order to burst by distinct column values *it is mandatory that the last sheet of the input report will be called ***burst**** and this sheet should contain the proper (meta)information required for bursting the report.

If *DocumentBurster* doesn't find the last sheet to have the name *burst* then it will fallback to process the report by using the previous alternative bursting method described in the section called "Burst by Distinct Sheets".

Note 2 - Replace All Excel Formulas With the Corresponding Calculated Values

In order to burst by distinct column values all the formulas (if the report contains formulas) should be pre-calculated before the report is sent for processing to *DocumentBuster*.

Microsoft Excel documentation - Replace a formula with its result [<http://office.microsoft.com/en-us/excel-help/replace-a-formula-with-its-result-HP010066258.aspx>]

Following screenshot shows how the (meta)information *burst* sheet should look.

| A | B | C | D | E | |
|------------------------|-----------------|------------------|----------------|---------------|------------|
| (*)burstMethod | burstSheetIndex | burstColumnIndex | (*)burstTokens | userVariables | configFile |
| distinct-column-values | -1 | -1 | Germany | | |
| | | | Mexico | | |
| | | | UK | | |
| | | | Sweden | | |
| | | | Germany | | |
| | | | France | | |
| | | | Spain | | |
| | | | Canada | | |
| | | | Argentina | | |
| | | | Switzerland | | |
| | | | Brazil | | |
| | | | Austria | | |
| | | | Italy | | |
| | | | Portugal | | |
| | | | USA | | |
| | | | Venezuela | | |
| | | | Ireland | | |
| | | | Belgium | | |
| | | | Norway | | |
| | | | Denmark | | |
| | | | Finland | | |
| | | | Poland | | |

(*)**burstMethod** : (distinct-sheets|distinct-column-values) - Required information which specifies the bursting method to be used. Following two valid values are possible

- *distinct-column-values* – should be used when splitting the report by the distinct values found in a column
- *distinct-sheets* – should be used when splitting the input report by each distinct sheet

Default value is *distinct-column-values*.

(*)**burstMethod** is an information which is mandatory to be provided.

burstSheetIndex - 0 (zero) based index of the sheet which needs to be split. For example, if it is required to split the first sheet then the index should be 0, if second sheet should be split then the index should be 1 and so on.

-1 is an allowed conventional value which means to split the first sheet of the Excel report.

Default value is -1.

burstColumnIndex - 0 (zero) based index of the column to be used when splitting the report. If it is required to split using the first column then *burstColumnIndex* should be defined as 0, if second column is required for splitting then *burstColumnIndex* should be defined as 1 and so on.

Usually the splitting column will be the last column from the splitting sheet. -1 is an allowed conventional value which means to split using the last column from the splitting sheet.

Default value is -1.

(*)burstTokens - required information which specifies the list of distinct burst tokens for which *DocumentBurster* should generate a separate output file. For example, in the above screenshot the list of burst tokens contains the set of distinct countries for which *DocumentBurster* should generate an output file.


()burstTokens* is an information which is mandatory to be provided.

userVariables - information to be provided in case there is a need to specify custom user variables for each distinct burst token. For more details on how to define user variables when bursting Excel reports please read Chapter 5, *Variables* .

configFile - information to be provided in case there is a need to process distinct Excel report types by using a different set of program configurations. By default, if no custom *configFile* is provided, *DocumentBurster* is processing all Excel reports using the default configuration file `./config/burst/settings.xml` . This should be fine in most situations, however there are cases when different reports will require different sets of configurations. For more details about how to process Excel reports using a custom configuration file please read the section called “Process Excel Reports Using a Custom Configuration File ” .

Next screenshot shows how the report `samples/Customers-Distinct-Column-Values.xls` was configured in order to be processed by *DocumentBurster*. For reasons of better visualization, the last column which is only used in this sample report for the purpose of report bursting, it is highlighted with the yellow color. It is not mandatory that the bursting column should have a specific color. Usually the burst column will be a hidden Excel column which will not be visible to the normal report users but only defined for the purpose of bursting the report.

(1) - *DocumentBurster* will keep the "empty rows" in the output reports. This is useful for keeping in the output files various headers and/or footers which a report might have.

| | A | B | C | D | E |
|----|------------------------------------|--------------|----------------|---|-------------|
| 1 | Customer List | | |  | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | 1 |
| 5 | Customer Name | City | Country | Contact Name | |
| 6 | Alfreds Futterkiste | Berlin | Germany | Maria Anders | Germany |
| 7 | Ana Trujillo Emparedados y helado | México D.F. | Mexico | Ana Trujillo | Mexico |
| 8 | Antonio Moreno Taquería | México D.F. | Mexico | Antonio Moreno | Mexico |
| 9 | Around the Horn | London | UK | Thomas Hardy | UK |
| 10 | Berglunds snabbköp | Luleå | Sweden | Christina Berglund | Sweden |
| 11 | Blauer See Delikatessen | Mannheim | Germany | Hanna Moos | Germany |
| 12 | Blondel père et fils | Strasbourg | France | Frédérique Citeaux | France |
| 13 | Bólide Comidas preparadas | Madrid | Spain | Martín Sommer | Spain |
| 14 | Bon app' | Marseille | France | Laurence Lebihan | France |
| 15 | Bottom-Dollar Markets | Tsawassen | Canada | Elizabeth Lincoln | Canada |
| 16 | B's Beverages | London | UK | Victoria Ashworth | UK |
| 17 | Cactus Comidas para llevar | Buenos Aires | Argentina | Patricio Simpson | Argentina |
| 18 | Centro comercial Moctezuma | México D.F. | Mexico | Francisco Chang | Mexico |
| 19 | Chop-suey Chinese | Bern | Switzerland | Yang Wang | Switzerland |
| 20 | Comércio Mineiro | São Paulo | Brazil | Pedro Afonso | Brazil |
| 21 | Consolidated Holdings | London | UK | Elizabeth Brown | UK |
| 22 | Die Wandernde Kuh | Stuttgart | Germany | Rita Müller | Germany |
| 23 | Drachenblut Delikatessen | Aachen | Germany | Sven Ottlieb | Germany |
| 24 | Du monde entier | Nantes | France | Janine Labrune | France |
| 25 | Eastern Connection | London | UK | Ann Devon | UK |
| 26 | Ernst Handel | Graz | Austria | Roland Mendel | Austria |
| 27 | Familia Arquibaldo | São Paulo | Brazil | Aria Cruz | Brazil |
| 28 | FISSA Fabrica Inter. Salchichas S. | Madrid | Spain | Diego Roel | Spain |
| 29 | Folies gourmandes | Lille | France | Martine Rancé | France |
| 30 | Folk och få HB | Bräcke | Sweden | Maria Larsson | Sweden |
| 31 | France restauration | Nantes | France | Carine Schmitt | France |
| 32 | Franchi S.p.A. | Torino | Italy | Paolo Accorti | Italy |
| 33 | Frankenversand | München | Germany | Peter Franken | Germany |
| 34 | Furia Bacalhau e Frutos do Mar | Lisboa | Portugal | Lino Rodriguez | Portugal |
| 35 | Galería del gastrónomo | Barcelona | Spain | Eduardo Saavedra | Spain |

(2) - Last column contains the values from which *DocumentBuster* will decide which Excel row goes to which output file. It can have any color and will usually be a hidden Excel column.

Copy the Header (and/or the Footer) of the Input Report To the Output Files

In real life most Excel reports will have a header and/or a footer (e.g. the company name/logo, the name of the report etc.) which is statically defined and which should be copied *ad litteram* to each of the output files.

| A | C | D | E |
|--------------------------|---------|--|---------|
| Customer List | |  | |
| | | | |
| | | | |
| | | | |
| Customer Name | Country | Contact Name | |
| Alfreds Futterkiste | Germany | Maria Anders | Germany |
| Ana Trujillo Emparedados | Mexico | Ana Trujillo | Mexico |
| Antonio Moreno Taquería | Mexico | Antonio Moreno | Mexico |
| Around the Horn | UK | Thomas Hardy | UK |
| Berglunds snabbköp | Sweden | Christina Berglund | Sweden |

(1) - *DocumentBuster* keeps the "empty rows" in the output reports. This is used to copy the static report headers or footers (and any other static row) to the output reports.

Steps to Follow When Bursting an Excel Report by Distinct Column Values

Following general steps should be followed when bursting an Excel report by distinct column values

- **Step 1** - Identify the appropriate *burst token* for the type of report which is being burst. A *burst token* can be anything which is uniquely identifying the document to be extracted such as the invoice ID, customer number or the email address where the document should be distributed.
- **Step 2** - Configure and prepare the input Excel report to include the (meta)information (the *burst* sheet) required for bursting the document.
- **Step 3** - This step is required only if the report contains Excel formulas. Before the report is sent for processing to *DocumentBuster* all the formulas should be pre-calculated and replaced with the corresponding values. *Microsoft Excel documentation - Replace a formula with its result* [<http://office.microsoft.com/en-us/excel-help/replace-a-formula-with-its-result-HP010066258.aspx>]
- **Step 4** - Burst the report and check the output files.

Sample - Customers-Distinct-Column-Values.xls

The previous steps will be exemplified when bursting the same samples/Customers-Distinct-Column-Values.xls.

Step 1 - Identify the appropriate *burst token*.

The report to be burst contains a list of customers and each customer has a country of origin. The requirement is to break the report by country and each output file should contain the customers from the relevant country.

The burst token is identified as being the **Country** column.

Step 2 - Configure and prepare the input Excel report to include the (meta)information required for bursting the document.

In order to burst by distinct column values it is mandatory that the *last sheet* of the input report will be called ***burst*** and this sheet should contain the proper (meta)information required for bursting the report.

-1 is an allowed conventional value which means to split the first sheet of the Excel report

-1 is an allowed conventional value which means to split using the last column from the splitting sheet

| A | B | C | D | E | |
|------------------------|-----------------|------------------|----------------|---------------|------------|
| (*)burstMethod | burstSheetIndex | burstColumnIndex | (*)burstTokens | userVariables | configFile |
| distinct-column-values | -1 | -1 | Germany | | |
| | | | Mexico | | |
| | | | UK | | |
| | | | Sweden | | |
| | | | Germany | | |
| | | | France | | |
| | | | Spain | | |
| | | | Canada | | |
| | | | Argentina | | |
| | | | Switzerland | | |
| | | | Brazil | | |
| | | | Austria | | |
| | | | Italy | | |
| | | | Portugal | | |
| | | | USA | | |
| | | | Venezuela | | |
| | | | Ireland | | |
| | | | Belgium | | |
| | | | Norway | | |
| | | | Denmark | | |
| | | | Finland | | |
| | | | Poland | | |

Burst tokens are defined as the list of countries

The last column from the splitting sheet is yellow (for better visualization) and is added only for the purpose of bursting the report. The bursting column doesn't need to have a specific color and will usually be an invisible Excel column (so that the normal report user will not see it).

| A | C | D | E |
|--------------------------|---------|--|---------|
| Customer List | |  | |
| | | | |
| | | | |
| Customer Name | Country | Contact Name | |
| Alfreds Futterkiste | Germany | Maria Anders | Germany |
| Ana Trujillo Emparedados | Mexico | Ana Trujillo | Mexico |
| Antonio Moreno Taquería | Mexico | Antonio Moreno | Mexico |
| Around the Horn | UK | Thomas Hardy | UK |
| Berglunds snabbköp | Sweden | Christina Berglund | Sweden |

(1) - Extra column which is added for the purpose of splitting the report. It can be defined as a hidden Excel column so that it will be invisible to the normal report users.

Step 3 - Calculate and replace Excel formulas with the corresponding values

Customers-Distinct-Column-Values.xls sample report doesn't contain any Excel formula, so there is nothing to do for this report in regards with that.

Step 4 - Burst the report and check the output files.

Burst samples/Customers-Distinct-Column-Values.xls report and check the list of output files, one Excel file for each country with each output file containing the list of customers from the specific country.

Chapter 3. Distribute Reports

Overview

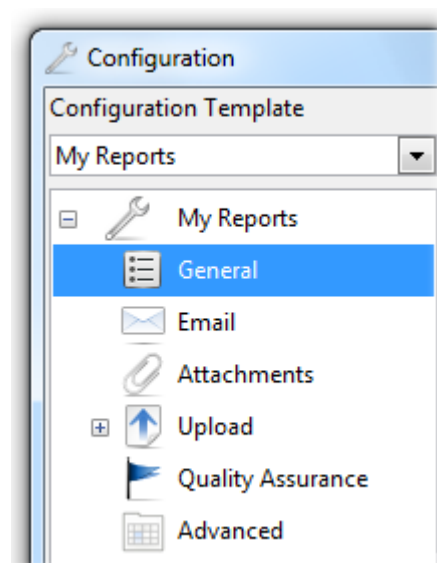
Your partners might need their reports to be sent through various distribution channels. Most of them will prefer email but others will rather use FTP. Some customers might have their own automated software systems to further process the reports and might expect the files to be placed in a specified remote location. Others might have more strict security related requirements and will prefer the FTPS protocol. In some scenarios the generated reports should be made directly available to the customers through an enterprise like portal, such as Microsoft SharePoint.

In order to support these scenarios, *DocumentBurster* offers a broad range of destination types like email, local files, FTP, FTPs, SFTP and TFTP, Windows shared drives, UNIX Samba servers and WebDAV servers. WebDAV protocol is used to distribute the reports to enterprise web portals such as Microsoft SharePoint, Oracle Portal or SAP NetWeaver.

Enable Report Distribution

In order to distribute the reports, the following configuration needs to be checked.

In *DocumentBurster* main window select Actions , Configure , General



| | |
|---|---|
| Burst File Name | <input type="text" value="\$burst_token\$.input_document_extension\$"/> |
| Default Merge File Name | <input type="text" value="merged.pdf"/> |
| Output Folder | <input type="text" value="output/.input_document_name\$/.now;format='yyyy.MM'"/> |
| Backup Folder | <input type="text" value="backup/.input_document_name\$/.now;format='yyyy.MM'"/> |
| Quarantine Folder | <input type="text" value="quarantine/.input_document_name\$/.now;format='yyyy."/> |
| <i>(should be a local folder)</i> | |
| Poll Folder | <input type="text" value="poll"/> |
| <i>(should be an existing local folder)</i> | |
| <input checked="" type="checkbox"/> | Distribute reports to Email, FTP ... |
| <input type="checkbox"/> | Delete reports once they are distributed |
| <input checked="" type="checkbox"/> | Quarantine reports which fail to be distributed |

Distribute reports to Email, FTP ... - If checked, the generated reports will be distributed as part of the bursting process, otherwise no. Default value is unchecked.

Distribute Reports by Email

Email Connection Settings

DocumentBuster can distribute reports using a Microsoft Exchange email server or using any other SMTP compliant email server. It is also possible to distribute reports using POP3 email servers like Hotmail, Yahoo! Mail, Gmail, etc.

In order to distribute reports by email, the email server connection settings should be properly configured.

In *DocumentBuster* main window select Actions , Configure , Email , Connection Settings

| | | |
|--------------------------------------|--|--|
| From Name | <input type="text" value="From Name"/> | <input type="button" value="Variables"/> |
| From Email Address | <input type="text" value="from@emailaddress.com"/> | <input type="button" value="Variables"/> |
| Host | <input type="text" value="Email Server Host"/> | <input type="button" value="Variables"/> |
| User Name | <input type="text" value="From Email User ID"/> | <input type="button" value="Variables"/> |
| Password | <input type="password" value="....."/> | <input type="button" value="Variables"/> |
| Port | <input type="text" value="25"/> | <input type="button" value="Variables"/> |
| <input type="checkbox"/> SSL Enabled | | |
| <input type="checkbox"/> TLS Enabled | | |

The email configuration is straightforward. The most important settings are the host, the user name, the password and the port. If an email server with SSL or TLS support (e.g. Gmail, etc.) is used then the appropriate checkboxes need to be selected.

Microsoft Exchange Server

If *DocumentBuster* is being used with a Microsoft Exchange email server, in this case, *DocumentBuster* should be configured with the same email connection settings as the ones already provided in the Microsoft Outlook email client software.

Hotmail, Yahoo! Mail, Gmail

These **public webmail providers** should be used **only** for sending **few emails**, for testing purposes. No matter which of these public services are used, you must be aware of the "rate limit" (the limit for sending emails) established by your email account provider.

For testing purposes, Gmail can be used as an email server for distributing few reports. If you are interested to integrate *DocumentBuster* with Gmail, *please read the following Gmail documentation*. [<http://mail.google.com/support/bin/answer.py?hl=en&answer=13287>]

Yahoo! Mail, Hotmail and other big email providers have POP3 support and can be configured with *DocumentBuster*. Please read the POP3 support documentation of the specific email provider which you have the intention to use.

Note 1

If required, a network or IT administrator from your organization should be able to give further help in configuring the email server settings.

Note 2

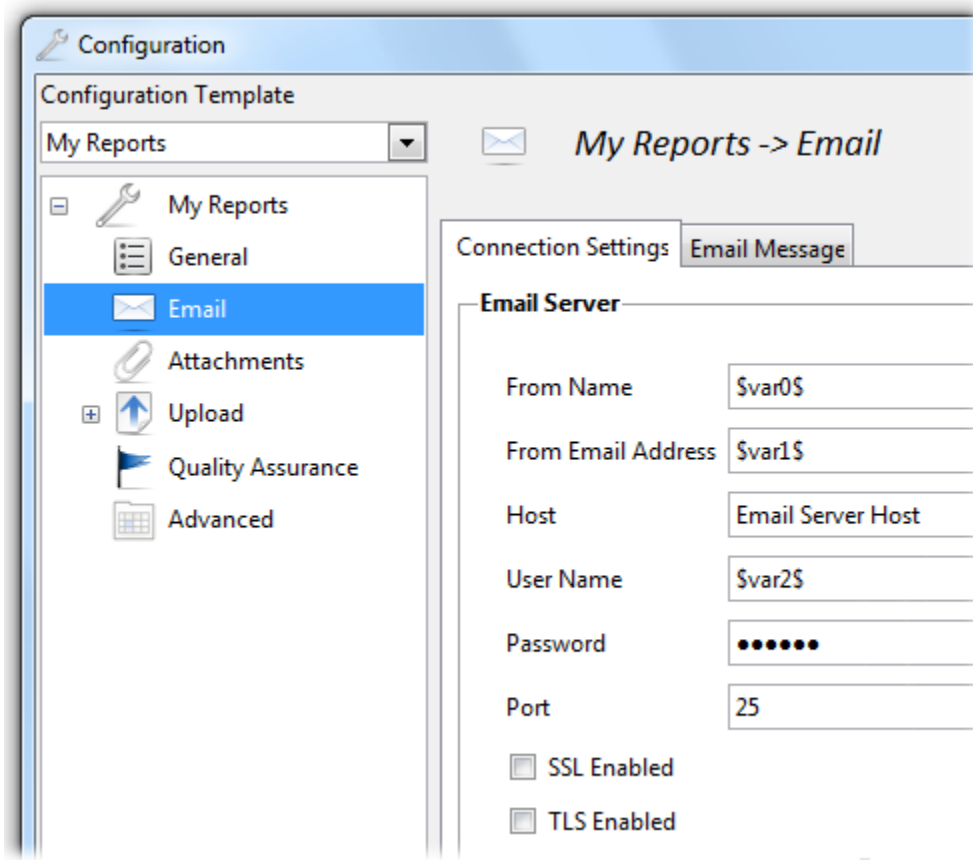
If there is a firewall or antivirus software which is enabled in-between *DocumentBuster* and the email server, in such case, the protecting software might need to be configured and allow *DocumentBuster* to send emails as a good and trustable citizen.

Dynamic Email Connection Settings

If required, using variables, the email connection settings can be dynamically filled at run-time when the reports are being distributed.

For example, your organization might have few departments (e.g. human resources/payroll, financial/accounting, travel department, etc.) and each department will need to distribute their own set of reports. Payslip reports (which are being distributed by the human resources/payroll department) should be sent using the *payroll@yourcompany.com* email account, invoices should be sent using *finance@yourcompany.com* and so on.

Variables, which are described with more details in Chapter 5, *Variables*, can be used to achieve such a requirement. Instead of hard-coding the sender email account as being *payroll@yourcompany.com* or *finance@yourcompany.com*, using variables, it is possible to dynamically fill the email connection details when the reports are being distributed (at that time the sender is known).



When sending the reports, the dynamic variables *\$var0\$*, *\$var1*, etc. will be appropriately evaluated with the configuration settings for either *payroll@yourcompany.com* or *finance@yourcompany.com*. For more details about *Variables* please check Chapter 5, *Variables*.

Email Each Report to One Recipient

The simplest email distribution scenario is when sending each of the generated reports to one different recipient.

By default the *To* address is configured with the value of the *\$burst_token\$* variable. If the burst tokens are of type email address, for example by using tokens such as *{clyde.grew@northridgehealth.org}*, *DocumentBuster* will send the output report to the corresponding email address, in this case the *clyde.grew@northridgehealth.org* email address.

This is how the existing sample report *samples/Payslips.pdf* is configured to work.

{clyde.grew@northridgehealth.org}

Northridge Pharmaceuticals
7649F Diamond Hts Blvd
San Francisco
(415) 872-9214

STATEMENT OF MONTHLY INCOME

| | |
|---------------|------------|
| Employee Name | Clyde Grew |
|---------------|------------|

using configuration

The screenshot shows the 'My Reports -> Email' configuration window. It has two tabs: 'General Settings' and 'Connection Settings'. The 'General Settings' tab is active, showing a 'Burst File Name' field with the value '\$burst_token\$.pdf'. The 'Connection Settings' tab is also visible, showing a 'Default Email Message' section with a 'To' field containing '\$burst_token\$'. Both the file name and the email address fields are highlighted with green boxes.

generates and emails to the
same email address

 clyde.grew@northridgehealth.org

The *burst token* is `clyde.grew@northridgehealth.org` so the output file name will be `clyde.grew@northridgehealth.org.pdf` and will be emailed as an attachment to the `clyde.grew@northridgehealth.org` email address.

Email Each Report to Multiple Recipients

Another scenario is to distribute each output report to a list of people (or groups of people). It is also easy to configure *DocumentBuster* to CC and BCC the generated reports.

The screenshot shows a configuration window titled "My Reports -> Email" with a contact email "support@pdfburst.com". It has two tabs: "Connection Settings" and "Email Message". The "Email Message" tab is active, showing fields for "To", "CC", and "BCC".

| Field | Value | Action |
|-------|---|-----------|
| To | emma.w@company.com;george.c@company.com | Variables |
| CC | cfo@company.com | Variables |
| BCC | finance@northridgehealth.com | Variables |

In addition, it is possible to use dynamic variables when configuring the email destinations for the output reports. For example, it is possible to configure *DocumentBuster* to distribute emails **TO** *\$var0\$*, **CC** to *\$var1\$* and **BCC** to *\$var2\$*.

This screenshot shows the same configuration window, but with dynamic variables entered in the email fields.

| Field | Value | Action |
|-------|----------|-----------|
| To | \$var0\$ | Variables |
| CC | \$var1\$ | Variables |
| BCC | \$var2\$ | Variables |

At run-time the variables will be expanded with values from the report which is being distributed, for example the variable values might be evaluated to

- *\$var0\$ (TO)*- emma.w@company.com;george.c@company.com
- *\$var1\$ (CC)*- accounting@company.com
- and *\$var2\$ (BCC)*- finance@northridgehealth.com.

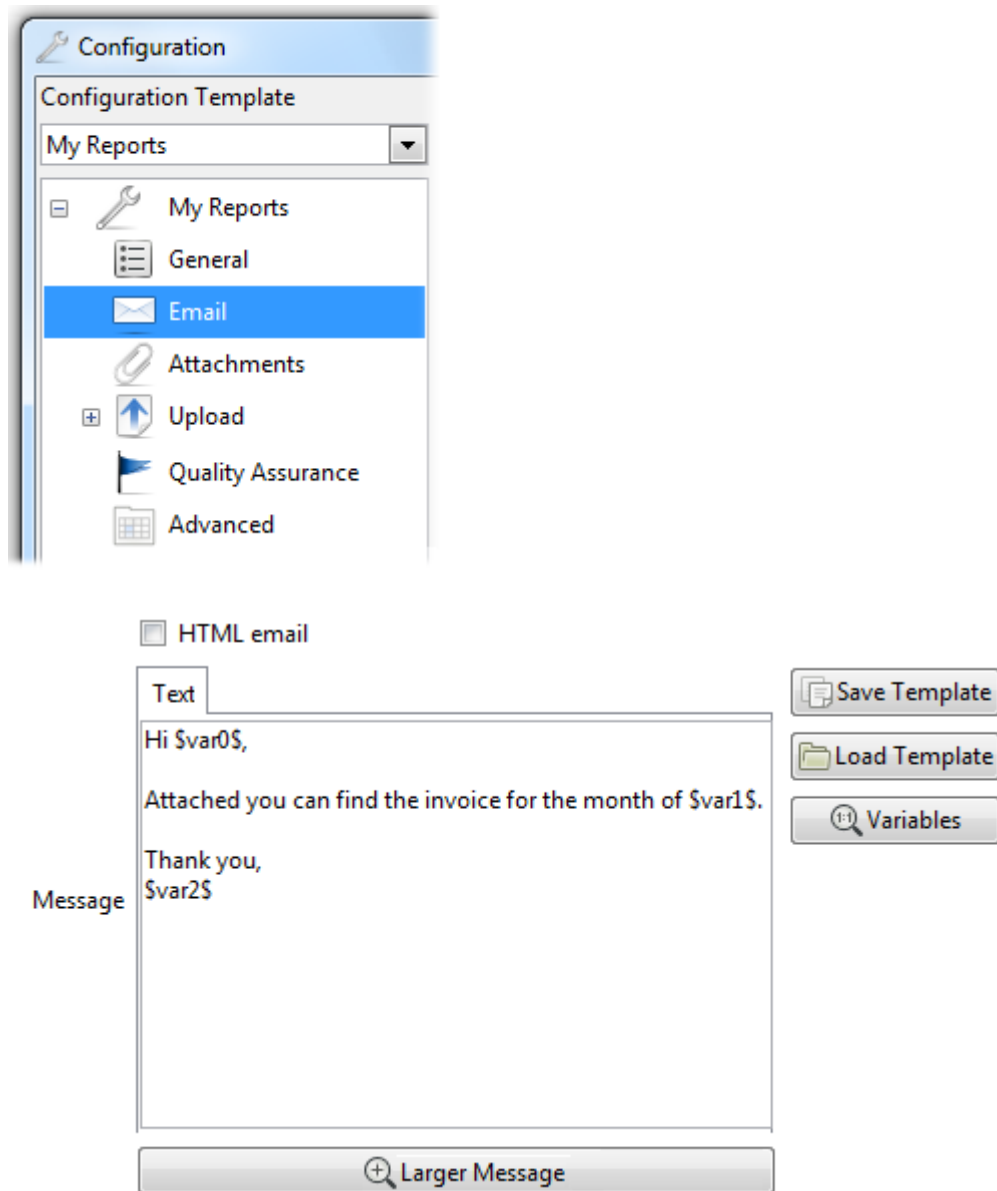
Note

Multiple email addresses should be separated by either semicolon (;) or comma (,).

Text Email Messages

DocumentBuster supports sending of configurable email messages which have the corresponding burst report attached to the email.

In *DocumentBuster* main window select Actions , Configure , Email , Email Message



Using variables, the subject and the text of the email messages can be dynamically configured and customized for each individual recipient.

Example - In the previous screenshot, when each individual report is being distributed, *\$var0\$*, *\$var1\$* and *\$var2\$* variables will be replaced with values fetched from the burst report such as *John*, *July* and *Michelle*.

Following is the message which is sent when the variables are expanded with the values *John*, *July* and *Michelle*

Hi John,

Attached you can find the invoice for the month of July.

Thank you,
Michelle

Note

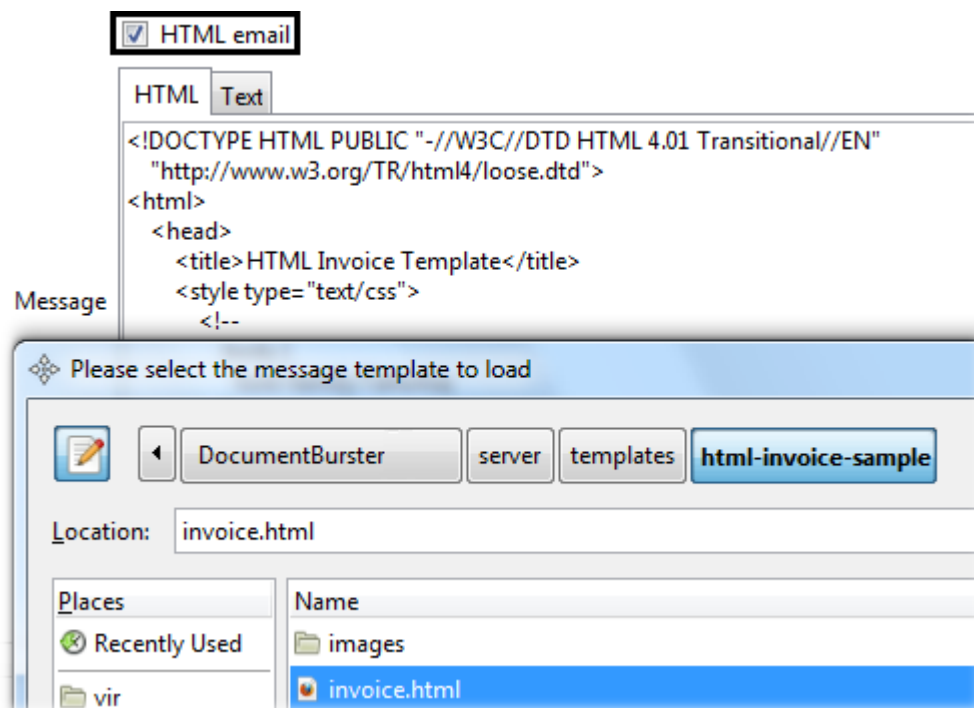
In order to avoid message retyping, using the *Save Template* and *Load Template* buttons, it is possible to save/load email message templates from/to external text files.

HTML Email Messages

DocumentBuster can be configured to send rich HTML formatted email messages.

DocumentBuster can send email messages with rich formatting in order to add color, images, headings, bulleted lists, emphasized text, underline key points, or to make some of the words bold.

In order to configure *DocumentBuster* to send emails with rich formatting, the *HTML email* checkbox should be selected and the HTML message should be defined with valid HTML code containing the message which needs to be distributed.



HTML Sample - invoice.html

DocumentBuster is coming with a sample HTML email template located in `templates/html-invoice-sample` folder. The sample template is called `invoice.html` and contains a good looking invoice to demonstrate the capabilities of HTML emails.

The sample HTML email template has a complex enough layout for giving an idea of what things can be achieved when sending HTML formatted emails (colors, images, bold text, etc.).

This is how the sample HTML invoice looks on Microsoft Outlook email client.

INVOICE

Northridge Pharmaceuticals
finance@northridgehealth.org

Invoice # 279
Created on 2008-10-09

Customer Details

Name: Alpha Electric
Email: accounting@alphaelectric.com.au
Payment Type: MasterCard

| Description | Qty | Unit Price | Amount |
|------------------|-----|------------|------------------|
| Nebulizer System | 10 | \$100.00 | \$1000.00 |
| Total | | | \$1000.00 |

Note 1

Every time an HTML email is configured, it is advisable to provide an alternative plain text message for HTML unaware email clients, such as text-based email clients.

Note 2

DocumentBurst resolves all the image paths used in the HTML code starting from `./templates` parent directory.

For example, the image `logo.png` is defined in the `invoice.html` having the relative path `src="html-invoice-sample/images/logo.png"`, starting from the `./templates` parent directory.

What happens when an image path is wrong

When *DocumentBurst* is not able to resolve an image (which has a wrong path defined in the HTML code), following things are happening.

- The corresponding burst report is not distributed by email

- If quarantine is configured, the corresponding burst report is copied to quarantine folder
- An exception is logged in the `logs/DocumentBurstster.log` log file. The log can be used to identify and fix the problematic image and path

Note 3

It is possible to use variables to dynamically customize the content of the HTML emails for each individual recipient.

Note 4

Before going to production, it is advisable to test the HTML email code through a large suite of email clients.

Attachments

Out of the box *DocumentBurstster* attaches to each email message the file which is extracted for the corresponding burst token. This is the most common scenario and it is supported using the default program configuration.

Sometimes a different situation might be required

- Send emails without any attachment
- Send emails with two or more files attached (separate files or a single archive/zip file containing all the files together)
- Distribute other document formats (e.g. Word, XML, etc.) beside the PDF and Excel formats which are supported out of the box

Use case example

The requirement is to send electronic invoices via email. The invoice is formed by a PDF which is the graphical form of the invoice, and an XML file which contains the details of the invoice.

By some countries law, it is required to send both files in order to consider this as a valid invoice.

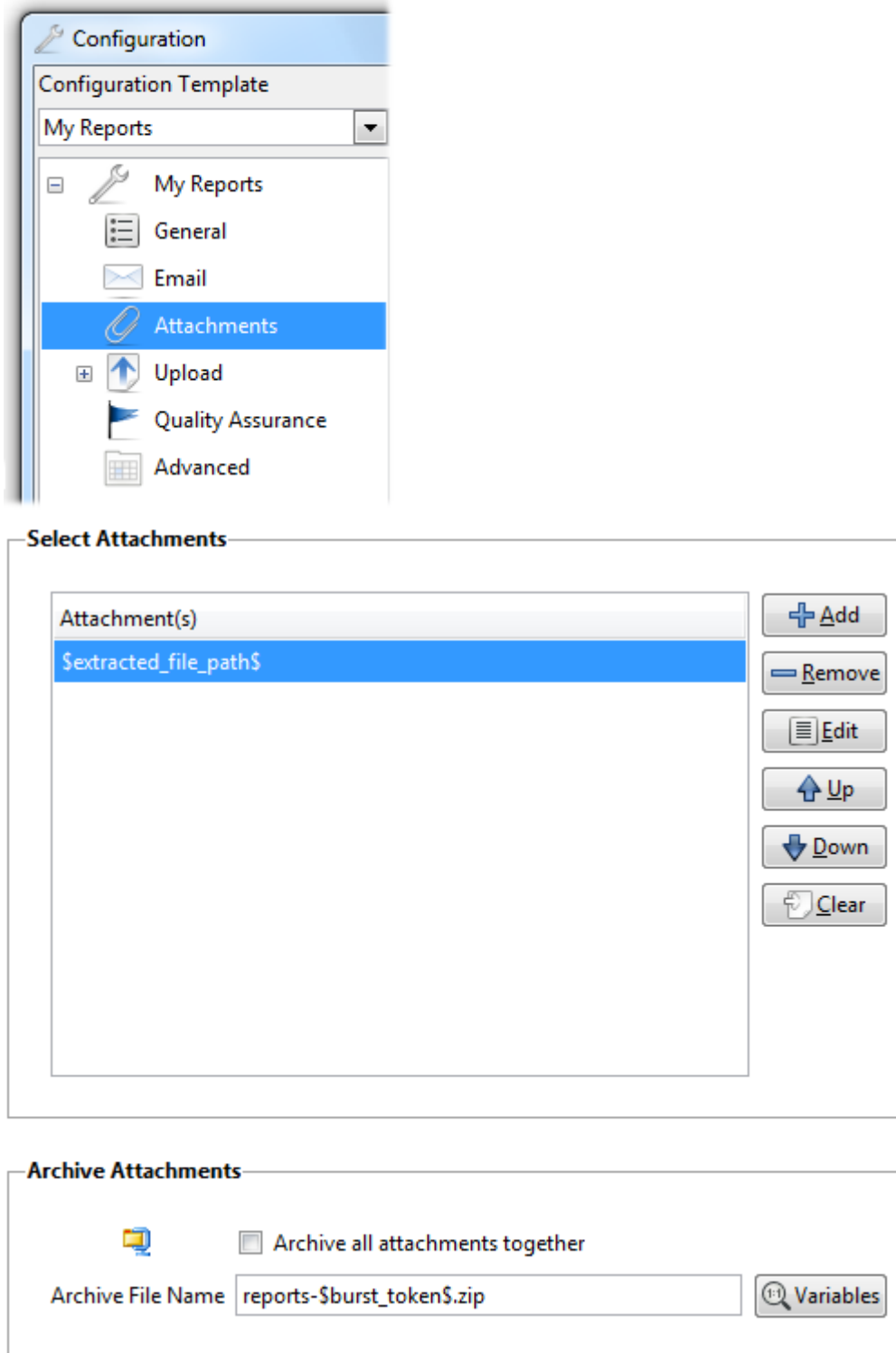
The above requirement can be achieved by defining both files as an attachment for the output emails

- The extracted PDF invoice
- The corresponding XML file

Furthermore, the files can be archived and sent together as a single zip file.

Configure Attachments

In *DocumentBurstster* main window select Actions , Configure , Attachments



The screenshot shows a 'Configuration' window with a 'Configuration Template' dropdown set to 'My Reports'. The left sidebar lists several configuration categories: 'My Reports' (with a sub-menu), 'General', 'Email', 'Attachments' (highlighted), 'Upload', 'Quality Assurance', and 'Advanced'. The 'Attachments' section is expanded, showing a list of attachment(s) with '\$extracted_file_path\$' selected. To the right of the list are buttons for '+ Add', '- Remove', 'Edit', 'Up', 'Down', and 'Clear'. Below this is the 'Archive Attachments' section, which includes a checkbox for 'Archive all attachments together' (unchecked), a text field for 'Archive File Name' containing 'reports-\$burst_token\$.zip', and a 'Variables' button.

Configuration

Configuration Template
My Reports

- My Reports
 - General
 - Email
 - Attachments**
 - Upload
 - Quality Assurance
 - Advanced

Select Attachments

| Attachment(s) | |
|-------------------------|---|
| \$extracted_file_path\$ | <input type="button" value="+ Add"/> |
| | <input type="button" value="- Remove"/> |
| | <input type="button" value="Edit"/> |
| | <input type="button" value="Up"/> |
| | <input type="button" value="Down"/> |
| | <input type="button" value="Clear"/> |

Archive Attachments

☐ Archive all attachments together

Archive File Name: reports-\$burst_token\$.zip

- By default the software is attaching *\$extracted_file_path\$*.
- Attachments can be added, removed and edited using the buttons with the corresponding names *Add*, *Remove* (or *Clear* to remove all) and *Edit*.

- *Variables* can be used when defining attachments. For more details about variables please read Chapter 5, *Variables*.
- The files are attached in the selection order. The order can be changed using *Up* and *Down* button commands.

Archive Attachments

- *Archive all attachments together* - If selected, *DocumentBuster* will archive (zip) all the attachments together. If configured to distribute reports, *DocumentBuster* will deliver to the relevant recipients the archived (zip) file containing all the corresponding files. Default value is unchecked.
- *Archive File Name* – Name for the archive file. Default value is *reports-\$burst_token\$.zip*.

Upload Reports

DocumentBuster software can upload business reports through most of the existing network protocols including FTP, Windows File Share, FTPS, SFTP/SCP/SSH, TFTP, HTTP, HTTPs and WebDAV. WebDAV is an extension of HTTP protocol and *DocumentBuster* can be used to upload reports to any web server which has WebDAV support.

cURL Integration

DocumentBuster closely integrates with *cURL*, a Swiss-army knife for doing data transfer. Through *cURL*, *DocumentBuster* can distribute reports via HTTP or FTP with or without authentication, it works over SSL, and it works without interaction. Actually *cURL* (and thus *DocumentBuster*) supports distributing files and data to a various range of common Internet protocols, currently including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, LDAP, LDAPS, DICT, TELNET, FILE, IMAP, POP3, SMTP and RTSP.

cURL - <http://curl.haxx.se/>

Cross platform

cURL is portable and works on many platforms, including Windows, Linux, Mac OS X, MS-DOS and more.

On Windows, *DocumentBuster* package distribution is bundling together a recent version of *cURL*. So, if your organization is running *DocumentBuster* under Windows, there is nothing more to download or install in regards with *cURL*.

For other UNIX like systems, such as Linux and Mac OS X, the appropriate *cURL* binaries distribution should be properly downloaded and installed. In addition, the *cURL* groovy scripts which are bundled together with *DocumentBuster* are written for Windows usage and should support small adjustments to be made ready for usage under Linux/UNIX.

Further *cURL* resources

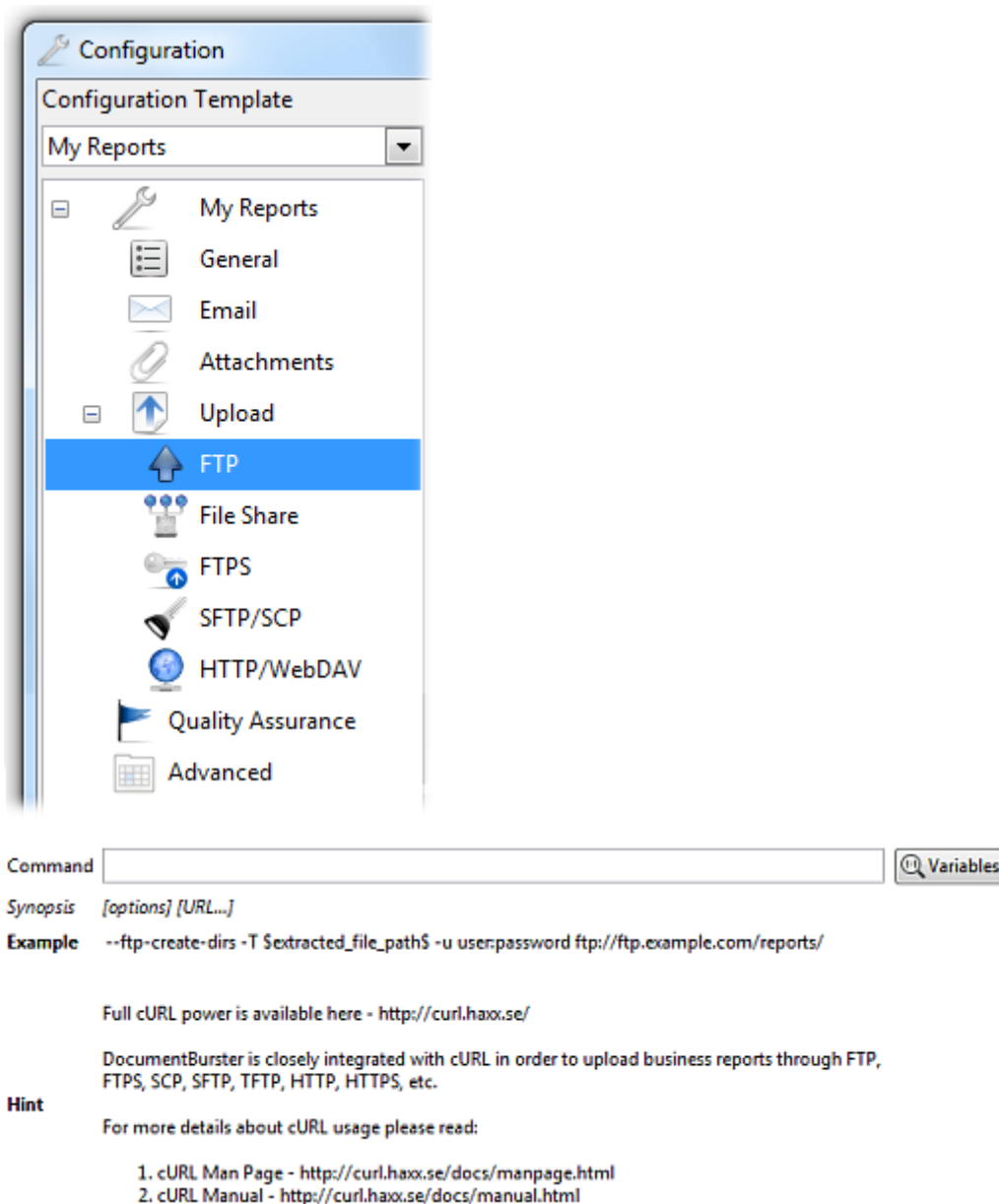
cURL is a tool for getting or sending files using URL syntax. The URL syntax is protocol-dependent. Along with the URL for the required protocol, *cURL* can take some additional options in the command line.

For complete *cURL* documentation you can follow

- *cURL Manual* [<http://curl.haxx.se/docs/manual.html>]
- *cURL Man Page* [<http://curl.haxx.se/docs/manpage.html>]
- *cURL Frequently Asked Questions* [<http://curl.haxx.se/docs/faq.html>]

FTP

In *DocumentBuster* main window select Actions , Configure , Upload , FTP



Command - Command which *DocumentBuster* will execute in order to upload each of the output reports. It is possible to use variables in order to dynamically generate the command at run-time.

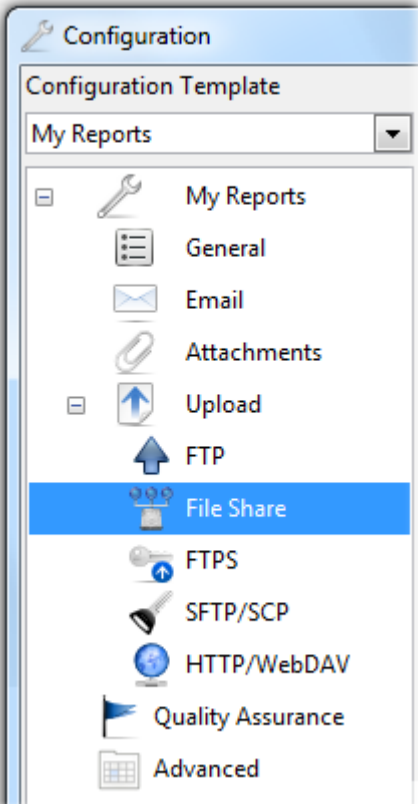
The example provided with the software is a typical cURL command for uploading reports using the FTP protocol.

- `--ftp-create-dirs` - (FTP/SFTP) When an FTP or SFTP URL/operation uses a path that doesn't currently exist on the server, the standard behavior is to fail. Using this option, the software will instead attempt to create missing directories.
- `-T, --upload-file` - This transfers the specified local file to the remote host.

- `$extracted_file_path$` - Variable which is expanded with the path to the last extracted file.
- `-u, --user <user:password>` - Specify the user name and password to use for server authentication. User variables can be used in order to dynamically configure user and password.
- `example.com` - Replace this with the FTP host where the reports should be uploaded. User variables can be used in order to dynamically configure the FTP host.
- `/reports/` - Replace this with the folder on the server where the reports should be uploaded. User variables can be used in order to dynamically configure the remote folder.

File Share

In *DocumentBuster* main window select Actions , Configure , Upload , File Share



The screenshot shows the 'Configuration' window in DocumentBuster. The 'Configuration Template' dropdown is set to 'My Reports'. The left sidebar shows a tree view with 'File Share' selected and highlighted in blue. Other options in the tree include 'My Reports', 'General', 'Email', 'Attachments', 'Upload', 'FTP', 'FTPS', 'SFTP/SCP', 'HTTP/WebDAV', 'Quality Assurance', and 'Advanced'.

Below the configuration tree, there is a 'Command' input field and a 'Variables' button. The 'Synopsis' section shows the command syntax: `[options] [URL...]`. The 'Example' section shows the command: `-T $extracted_file_path$ file://hostname/path/to/the%20folder`.

Use Case Examples:

1. Publish, archive or file reports to corporate network locations
2. Publish, archive or file reports to (web) portals (e.g. Microsoft SharePoint, etc.)

Hint

For more details about cURL file:// usage please read:

1. cURL Man Page - <http://curl.haxx.se/docs/manpage.html>
2. cURL Manual - <http://curl.haxx.se/docs/manual.html>

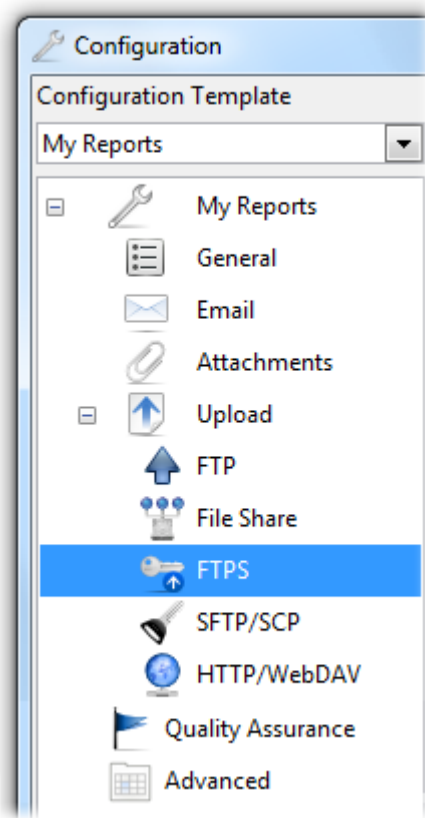
Command - Command which *DocumentBuster* will execute in order to upload each of the output reports. It is possible to use variables in order to dynamically generate the command at run-time.

The example provided with the software is a typical cURL command for uploading reports to file share.

- *file://* - *File URI scheme* [http://en.wikipedia.org/wiki/File_URI_scheme]
- *%20* - ASCII character code for space (URL Encoding).
- *\$extracted_file_path\$* - Variable which is expanded with the path to the last extracted file.

FTPS

In *DocumentBuster* main window select Actions , Configure , Upload , FTPS



Command

Variables

Synopsis *[options] [URL...]*

Example `--ssl -T $extracted_file_path$ -u user:password ftp://ftp.example.com/reports/`

FTPS (also known as FTP-ES, FTP-SSL and FTP Secure) is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols.

FTPS should not be confused with the SSH File Transfer Protocol (SFTP), an incompatible secure file transfer subsystem for the Secure Shell (SSH) protocol. It is also different from Secure FTP, the practice of tunneling FTP through an SSH connection.

Hint

Source: <http://en.wikipedia.org/wiki/FTPS>

For more details about cURL FTPS usage please read:

1. cURL Man Page - <http://curl.haxx.se/docs/manpage.html>
2. cURL Manual - <http://curl.haxx.se/docs/manual.html>

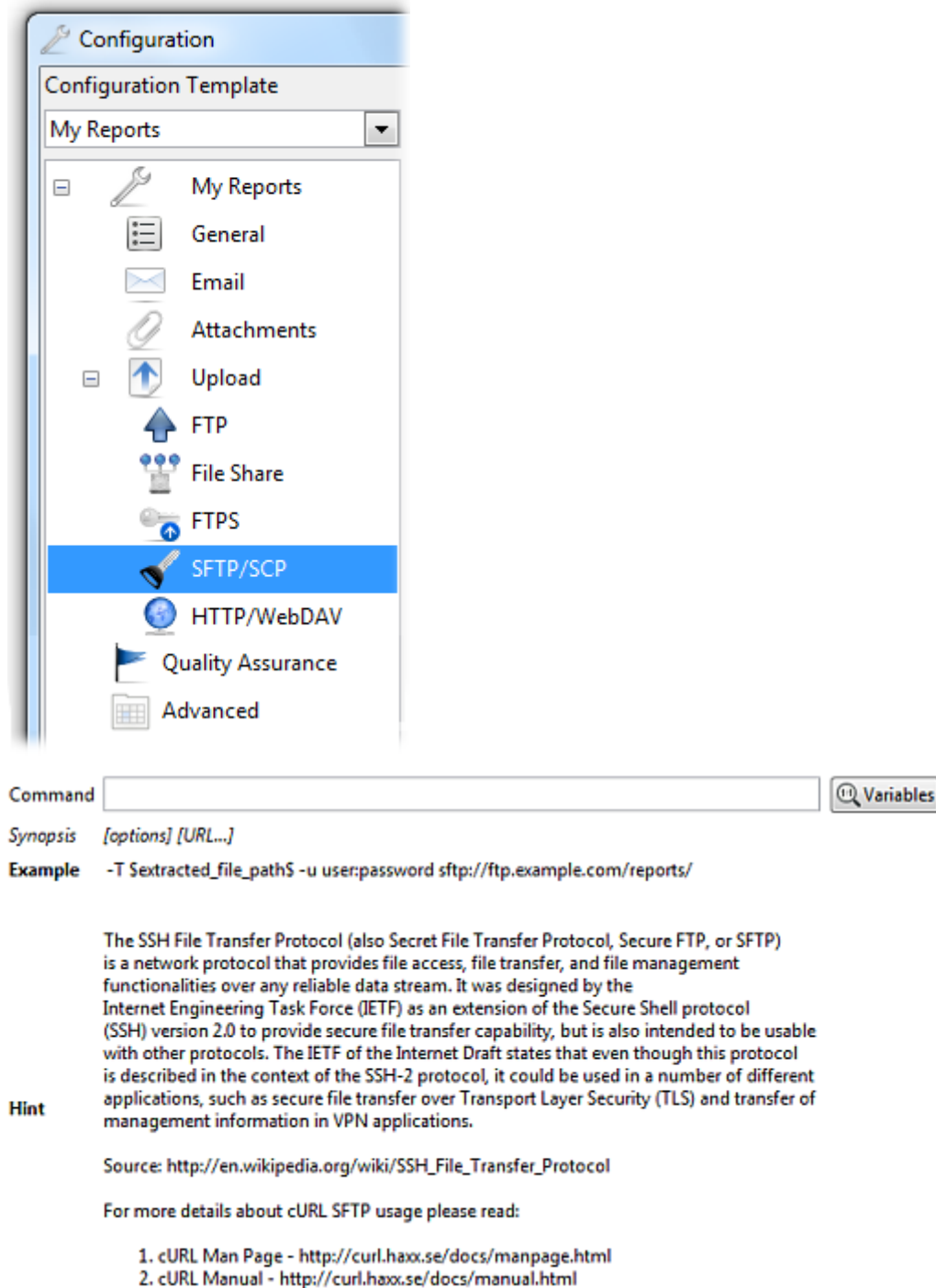
Command - Command which *DocumentBuster* will execute in order to upload each of the output reports. It is possible to use variables in order to dynamically generate the command at run-time.

The example provided with the software is a typical cURL command for uploading reports to a server which has FTPS support.

- `--ssl` - Try to use SSL/TLS for the connection. Reverts to a non-secure connection if the server doesn't support SSL/TLS. See also `--ftp-ssl-control` and `--ssl-reqd` for different levels of encryption required.
- `$extracted_file_path$` - Variable which is expanded with the path to the last extracted file.

SFTP/SSH/SCP

In *DocumentBuster* main window select Actions , Configure , Upload , SFTP/SCP



Configuration

Configuration Template

My Reports

- My Reports
 - General
 - Email
 - Attachments
 - Upload
 - FTP
 - File Share
 - FTPS
 - SFTP/SCP**
 - HTTP/WebDAV
 - Quality Assurance
 - Advanced

Command Variables

Synopsis [options] [URL...]

Example -T \$extracted_file_path\$ -u user:password sftp://ftp.example.com/reports/

Hint

The SSH File Transfer Protocol (also Secret File Transfer Protocol, Secure FTP, or SFTP) is a network protocol that provides file access, file transfer, and file management functionalities over any reliable data stream. It was designed by the Internet Engineering Task Force (IETF) as an extension of the Secure Shell protocol (SSH) version 2.0 to provide secure file transfer capability, but is also intended to be usable with other protocols. The IETF of the Internet Draft states that even though this protocol is described in the context of the SSH-2 protocol, it could be used in a number of different applications, such as secure file transfer over Transport Layer Security (TLS) and transfer of management information in VPN applications.

Source: http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol

For more details about cURL SFTP usage please read:

1. cURL Man Page - <http://curl.haxx.se/docs/manpage.html>
2. cURL Manual - <http://curl.haxx.se/docs/manual.html>

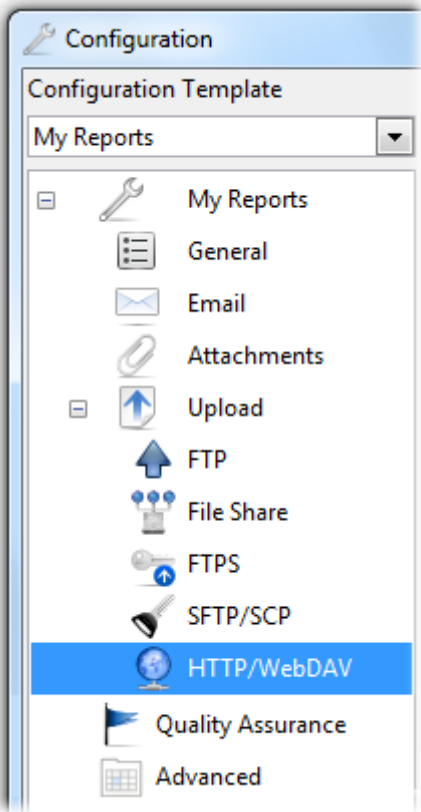
Command - Command which *DocumentBuster* will execute in order to upload each of the output reports. It is possible to use variables in order to dynamically generate the command at run-time.

The example provided with the software is a typical cURL command for uploading reports to a server which has SFTP support.

- *sftp://* - Specifies that SFTP protocol should be used.
- *\$extracted_file_path\$* - Variable which is expanded with the path to the last extracted file.

HTTP/WebDAV

In *DocumentBurst* main window select Actions , Configure , Upload , HTTP/WebDAV



The screenshot shows the 'Configuration' dialog box in DocumentBurst. The 'Configuration Template' dropdown is set to 'My Reports'. The left sidebar lists various configuration categories: My Reports, General, Email, Attachments, Upload, FTP, File Share, FTPS, SFTP/SCP, HTTP/WebDAV (highlighted in blue), Quality Assurance, and Advanced.

Below the sidebar, the 'Command' field is empty, and a 'Variables' button is visible. The 'Synopsis' section shows the command syntax: `[options] [URL...]`. The 'Example' section provides a command: `-T $extracted_file_path$ http://www.example.com/`. The 'SharePoint Example' section provides a command: `--ntlm -T $extracted_file_path$ -u user:password https://sharepointserver.com/reports/`.

Use Case Examples:

1. Publish, archive or file reports to (web) portals like Microsoft SharePoint, IBM WebSphere Portal, Oracle Portal, SAP NetWeaver, Tibco PortalBuilder, Samsung ACUBE Portal or to any of the major open source portal applications such as Liferay Portal, Hippo portal, JBoss Enterprise Portal, eXo, Apache Portal, etc.
2. Publish, archive or file reports to document management systems such as EMC Documentum, OpenText ECM, Alfresco, etc.

Hint

For more details about cURL HTTP/WebDAV usage please read:

1. cURL Man Page - <http://curl.haxx.se/docs/manpage.html>
2. cURL Manual - <http://curl.haxx.se/docs/manual.html>

Command - Command which *DocumentBurst* will execute in order to upload each of the output reports. It is possible to use variables in order to dynamically generate the command at run-time.

The example provided with the software is a typical cURL command for uploading reports to a web server. The web server should have HTTP upload enabled, otherwise uploads will not be allowed.

Microsoft SharePoint is a common report distribution use case. The command to upload to a SharePoint portal should be similar with the provided SharePoint example.

Chapter 4. Configuration Templates

If required, *DocumentBuster* has support to use a different set of configuration settings for each separate report type. For example, using this capability, it is possible to use a different email subject, text or burst file name for *invoices* report type and a different email subject, text or burst file name when processing *payslips* report type.

By default, if no custom configuration file is provided, *DocumentBuster* will process all the reports using the default configuration settings which are saved in `./config/burst/settings.xml` configuration file. This should be fine in most of the situations, however there are cases when it is required to process a different report type using its own set of configuration settings.

Use case example

The requirement is to burst and email to customers several different types of financial documents, specifically

- invoices
- statements
- dunning letters

Each of the above report types should be delivered to customers using an email subject and text which is specific to the report type which is being distributed (i.e. invoices, statements or dunning letters).

The requirement can be achieved by defining

- *invoices* email subject and text in a configuration file such as `./config/burst/invoices.xml`
- *statements* email subject and text in a different `./config/burst/statements.xml`
- and *dunning letters* email subject and text in its own configuration file `./config/burst/dunning-letters.xml`.

DocumentBuster will process and email each different report type by using its own distinct set of configuration settings.

Process PDF Reports Using a Custom Configuration File

While bursting a PDF report, *DocumentBuster* checks the first page of the report for a custom defined configuration file by looking for a pattern like

```
<config>path-to-the-custom-settings-file.xml</config>
```

For example, in order to process *invoices* report using a custom configuration file, the first page of the input `invoices.pdf` report should contain something similar with

```
<config>./config/burst/invoices.xml</config>
```

Note

The same result can be achieved by using the shorter (while being less expressive) pattern

```
<c>path-to-the-custom-settings.xml</c>,
```

for example `<c>./config/burst/invoices.xml</c>`

Process Excel Reports Using a Custom Configuration File

While bursting an Excel report, *DocumentBuster* checks for a custom configuration file defined in *configFile* column from *burst* metadata sheet. If no custom configuration file is found then *DocumentBuster* will burst the report using the default `./config/burst/settings.xml` configuration file.

Following is how an Excel reports should be configured in order to achieve the above (i.e. invoices, statements and dunning letters) requirement.

In *burst* sheet, *invoices.xls* Excel report should have the custom *configFile* defined like that

configFile

`./config/burst/invoices.xml`

In *burst* sheet, *statements.xls* Excel report should have the custom *configFile* defined like that

configFile

`./config/burst/statements.xml`

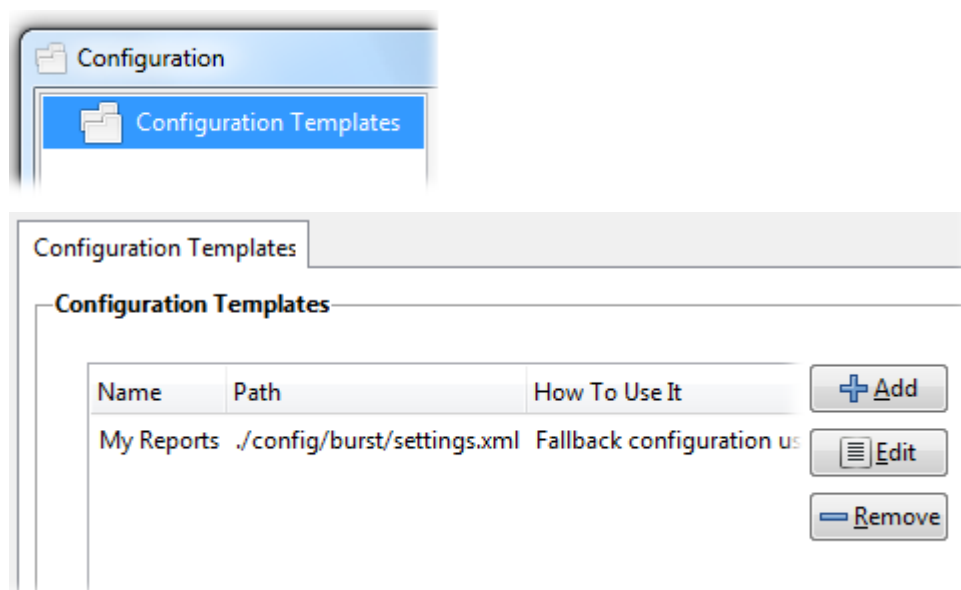
In *burst* sheet, *dunning-letters.xls* Excel report should have the custom *configFile* defined like that

configFile

`./config/burst/dunning-letters.xml`

Add, Edit and Delete Configuration Templates

In *DocumentBuster* main window select Actions , Configuration Templates



Name

(e.g. Payslips, Invoices, Statements, etc.)

Copy Initial Values From

Path

How To Use It

- *Add* - Create a new configuration template.
- *Edit* - Modify an existing configuration template.
- *Remove* - Delete an existing configuration template.

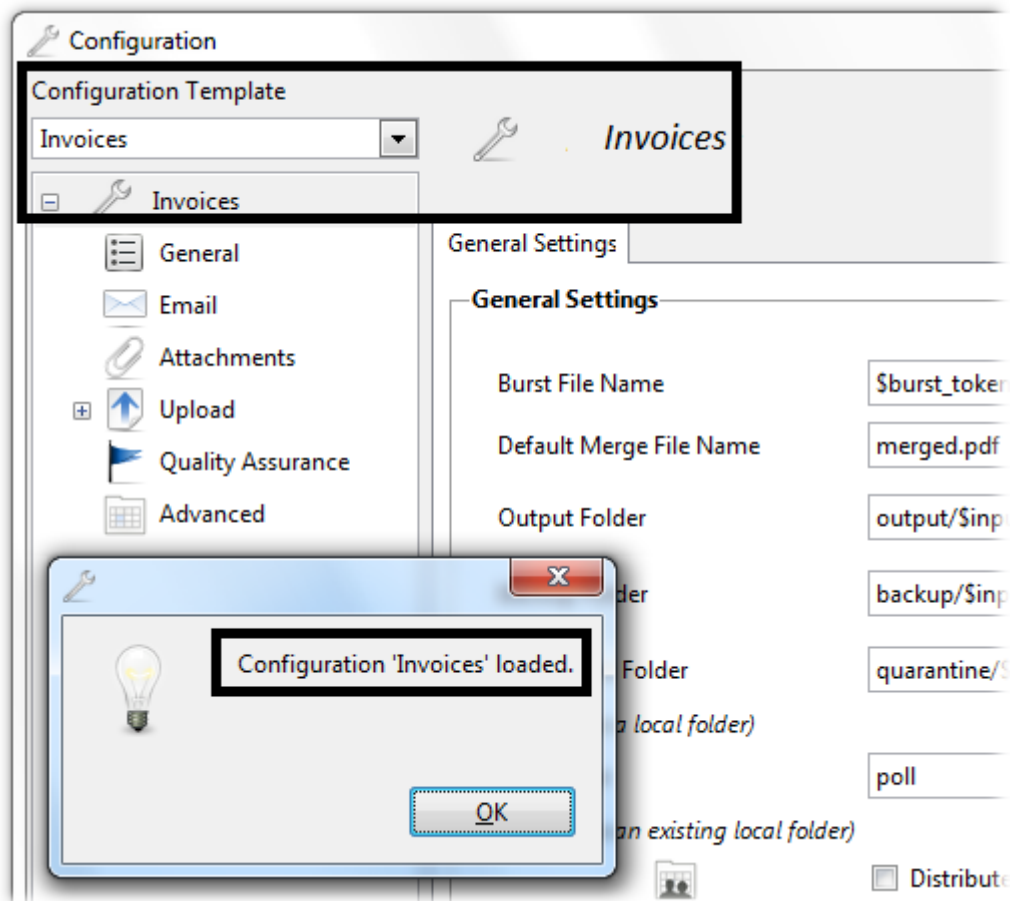
Each configuration template is saved in a different configuration file. For example, *Invoices* configuration template might be saved in a `./config/burst/invoices.xml` file while *Statements* configuration template might be saved in a `./config/burst/statements.xml` file.

Create Similar Configuration Templates

Copy Initial Values From - When creating a new configuration template it is required to provide the path to an existing and valid configuration file from which the initial configuration values will be copied. Once the new configuration template is created you might modify only the few settings which should be different between configuration templates (e.g. email subject, email message, etc.). Most of the other settings will usually remain the same (e.g. email server connection settings) between different configuration templates.

Default Value - `./config/burst/default/defaults.xml`

Once the configuration templates are created they will become available to be used throughout the *DocumentBuster* GUI.



Chapter 5. Variables

DocumentBuster variables are pieces of information from the input report, evaluated and processed individually for each output report.

Variables can be used to define dynamic configurations as well as data to be included in the delivery of documents (e.g. using variables and an email template it is possible to define a personalized email subject/message for each individual recipient).

Variables can be used to define custom dynamic values for the following configurations

- *Burst File Name*
- *Output Folder*
- *Backup Folder*
- *Quarantine Folder*
- *Upload (FTP, File Share, etc.) commands and URLs* can be dynamically generated
- Email *To*, *CC* and *BCC* fields can be dynamically generated
- *Email subject and message* text can be personalized using variables
- Email *connection settings* can be customized using variables. *From Name*, *From Email Address*, *Host*, *User Name*, *User Password* and *Port* settings are all configurable using variables.

Using variables, the values of the above configurations can be dynamically populated at run-time with information coming from the report which is being burst.

Example - Configure an Unique Output Folder

While it is possible to define static values for the output folders, it is not advisable. In order to avoid the output reports to get overridden from one burst session to another, with the help of variables, it is better to define dynamic *timestamp based* output folder configurations.

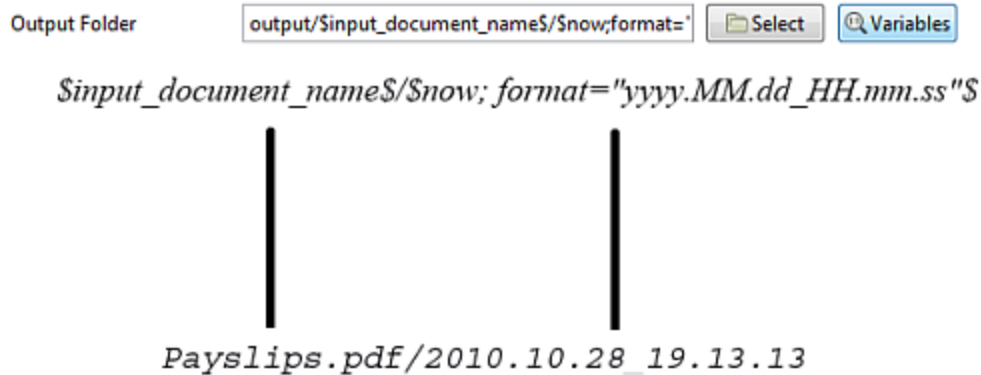
Following are few situations in which variables will help

- The same report is being burst at different times. Bursting the same report to the same statically defined output folders will override the files generated during previous burst sessions.
- Few different reports are using the same burst tokens (for example the email address of the same client). Having a common output folder will result in getting the generated reports to be overridden between different burst sessions. This is happening because the same burst token is found in different input reports.

Using variables, with unique *timestamp based* values generated at run-time, it is possible to overcome the above described problems by defining unique output folder names per each different burst session.

Output Folder, *Backup Folder* and *Quarantine Folder* configurations are all defined using the same pattern `$input_document_name$/$now; format="yyyy.MM.dd_HH.mm.ss"$`

When bursting the input sample document *Payslips.pdf*, by default, the output files are being generated in a folder similar with *Payslips.pdf/2010.10.28_19.13.13* - `$input_document_name$/$now; format="yyyy.MM.dd_HH.mm.ss"$`.



Built-In vs. User-Defined Variables

DocumentBurster has two types of variables

- Built-In Variables
- User-Defined Variables

Built-In Variables

Built-in variables can be the name of the report to burst, the date (in various formats) when the bursting is happening or the current burst token.

Following built-in variables are available to be used within *DocumentBurster*

- `$input_document_name$` - the file name of the input report
- `$input_document_extension$` - the file extension of the input report. It will be *pdf* for PDF input reports, *xls* for Excel versions prior to 2007 and *xlsx* for Excel 2007 and Excel 2010.
- `$burst_token$` - the burst token which is used for bursting the current file
- `$burst_index$` - the index of the burst file. For example, a file which will be burst fourth will have the value 4
- `$output_folder$` - the output folder where the last file was extracted
- `$extracted_file_path$` - the path to the last extracted file
- `$now;format="yyyy.MM.dd_HH.mm.ss"$` - the current date and time displayed in the specified format. Custom date formats can be specified also, for example it is possible to display the full date/time or display one or a combination of year, month, week, day, hour, minute, second. *yyyy.MM.dd_HH.mm.ss* is the format which is provided with the software.

Note

Windows doesn't allow to use *the character* : for defining the folder and file names.

Full date format documentation is available here [<http://download.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html>]

- *\$now_default_date\$* - shortcut to the default date format in the computer's locale settings. U.S. Locale example would be Jun 30, 2009
- *\$now_short_date\$* - shortcut to the short date format in the computer's locale settings. U.S. Locale example would be 6/30/09
- *\$now_medium_date\$* - shortcut to the medium date format in the computer's locale settings. U.S. Locale example would be Jun 30, 2009
- *\$now_long_date\$* - shortcut to the long date format in the computer's locale settings. U.S. Locale example would be June 30, 2009
- *\$now_full_date\$* - shortcut to the full date format in the computer's locale settings. U.S. Locale example would be Tuesday, June 30, 2009

Using built-in variables it is possible to build an advanced foldering and archiving solution for the output reports.

Following different foldering options are possible. Combinations of the following are also possible

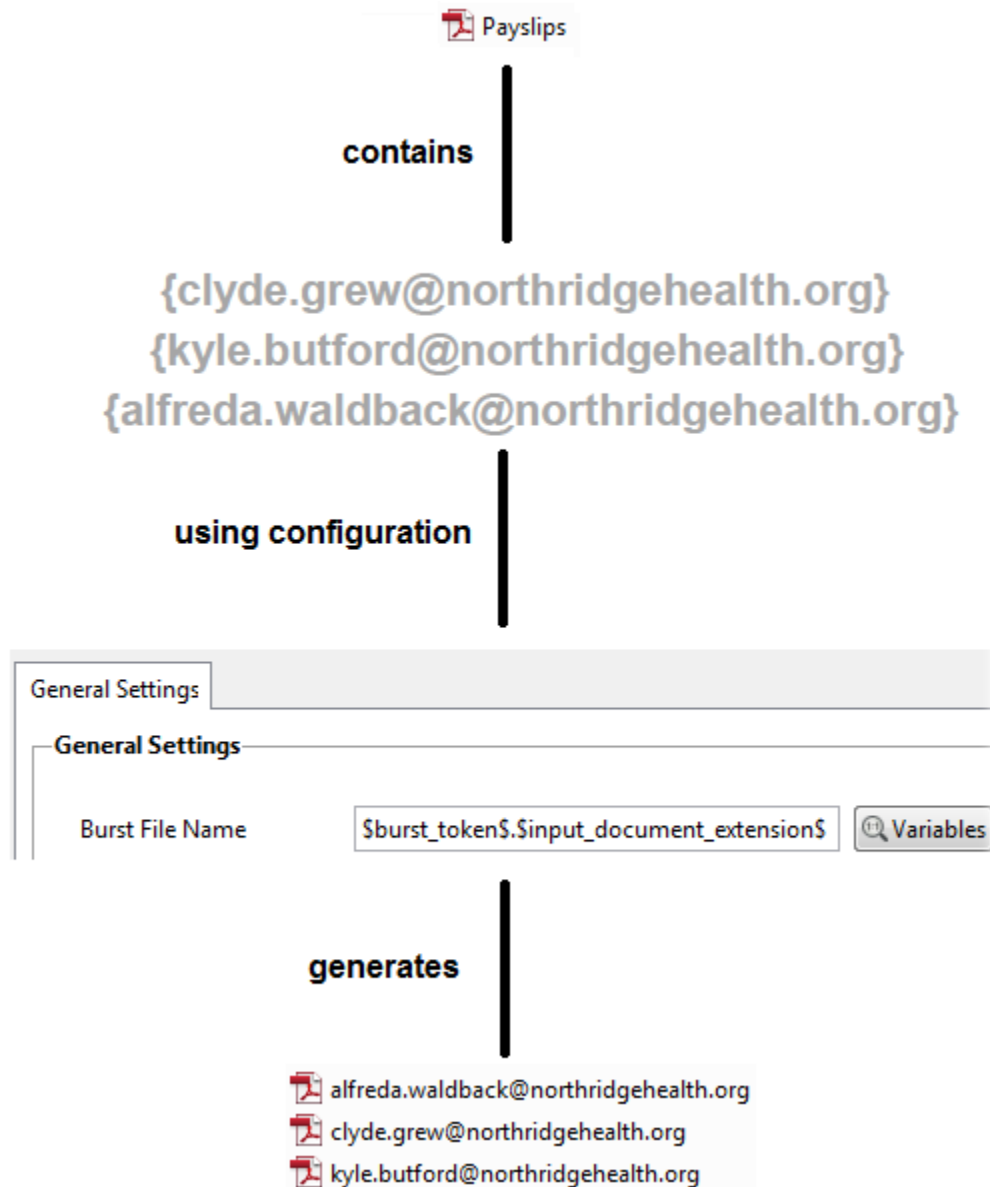
- Generate one different output folder per each input report
- Generate one different output folder per each different burst token
- Date related options - generate one output folder per year, financial quarter, month, week in month, day in week and up to the level of hours, minutes and seconds.

Sample - Payslips.pdf

DocumentBurst is coming with the following default settings

Burst File Name - *\$burst_token\$. \$input_document_extension\$*

When bursting the *samples/Payslips.pdf* report following three files are being generated. The three email addresses are the burst tokens and pdf is the input document extension.



User-Defined Variables

User defined variables can be any text from the report which is being burst or distributed.

User variables might be used for sending emails with a personalized subject and a personalized message text or for generating dynamic file names and folder names for the output burst reports.

DocumentBuster has support for up to 10 (ten) user defined variables *\$var0\$*, *\$var1\$*, *\$var2\$*, *\$var3\$*, *\$var4\$*, *\$var5\$*, *\$var6\$*, *\$var7\$*, *\$var8\$* and *\$var9\$*. While the variable names are not impressive they are for sure handy to use.

The values for the user defined variables are being populated with text content from the report which is being burst. Usually the variables will have different values for each different *burst token* (or for each different recipient).

In order to populate the user defined variables with values, *DocumentBuster* engine is looking inside the report for patterns like the following

- `<0>` any text which should be assigned as a value to the first variable `</0>` or
- `<1>` any text which should be assigned as a value to the second variable `</1>`

DocumentBuster engine supports up to 10 (ten) different variables, so the last variable will look like `<9>` any text which should be assigned as a value to the 10th variable `</9>`.

User defined variables can be used to dynamically generate any of the following *Burst File Name*, *Output Folder*, *Backup Folder*, *Quarantine Folder*, *Upload(FTP, File Share, etc.) commands and URLs*, *Email To*, *CC and BCC fields*, *Email Subject*, *Email Message Text*, *From Name*, *From Email Address*, *Host*, *User Name*, *User Password* and *Email Server Port*.

Sample Invoices-Oct.pdf - Customizable Burst File Name

With the default settings, *DocumentBuster* is generating the output file names using the following configuration

Burst File Name - `$burst_token$. $input_document_extension$`

where `$burst_token$` is the system built-in variable used to burst the separate files and `$input_document_extension$` is the input file extension (i.e. pdf, xls or xlsx).

Requirement

For example, a requirement might be to generate the output file names (*Burst File Name*) using the following pattern

Customer name-Invoice number-Invoice date.pdf

The requirement can be achieved with the help of user defined variables. Please check `samples/Invoices-Oct.pdf` sample report to see that *Invoice number* field is configured as burst token and *Customer name* and *Invoice date* fields are mapped to `$var1$` and `$var0$` user variables.

INVOICE

Northridge Pharmaceuticals

7649F Diamond Hts Blvd
San Francisco, CA 91324
USA
(415) 872-9214

info@northridgehealth.org
<http://www.northridgehealth.org>

| | |
|----------------|-----------------------|
| Invoice number | {0011} |
| Invoice date | <0>Oct 10, 2011</0> |
| Payment terms | Due on date specified |
| Due date | Oct 17, 2011 |

Client Details

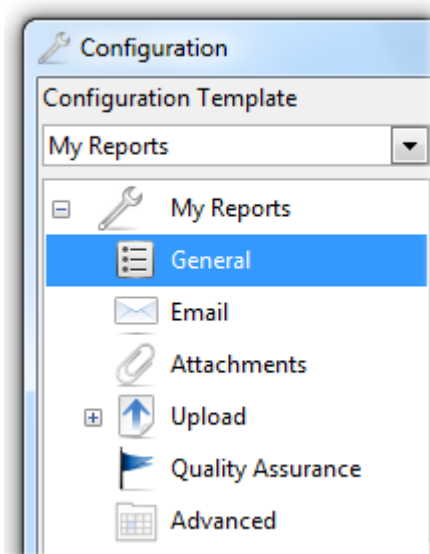
<1>Alpha Electric</1>

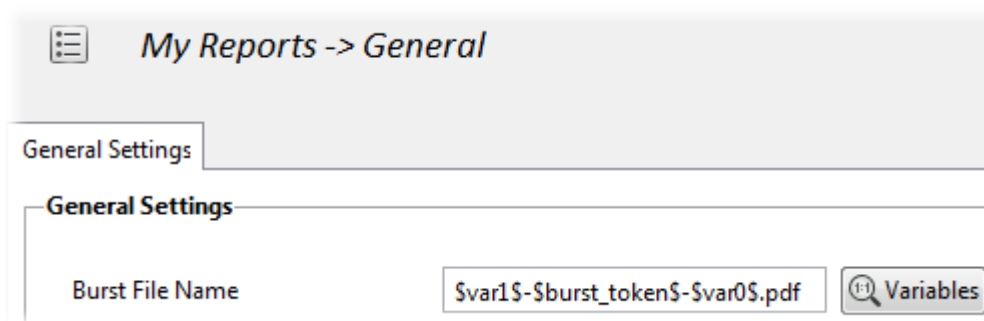
795E Driagram
Tucson, AZ 85705
USA

<2>accounting@alphaelectric.biz</2>

| Description | Quantity | Unit price | Amount |
|------------------|----------|--------------|------------------|
| Nebulizer system | 10 | \$100.00 | \$1000.00 |
| | | Total | \$1000.00 |

Burst File Name will be defined as \$var1\$-\$burst_token\$-\$var0\$.pdf

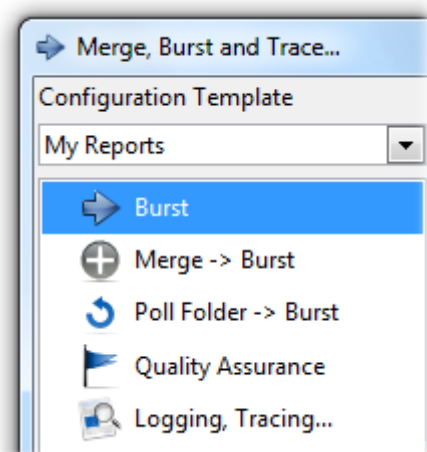







\$burst_token\$, \$var0\$ and \$var1\$ variables will be populated at run-time with values fetched from each separate report.


- Variables are fetched at run-time from each separate invoice. *DocumentBuster* is looking for `<N>value</N>` patterns in each invoice, where *N* is from 0 to 9.
- In the above example the name of the generated file will be Alpha Electric-0011-Oct 10, 2011.pdf


Using the previous configuration, bursting `samples/Invoices-Oct.pdf` will generate the following four distinct files





PDF/Excel File Path  **Select**

 **Burst**  **View Generated Reports**

 **Distribute reports to Email, FTP ...**


 **Delete reports once they are distributed**





 **Quarantine reports which fail to be distributed**

 **Are you sure you want to post another job?**

General Settings

General Settings

Burst File Name  **Variables**

 Alpha Electric-0011-Oct 10, 2011
 General Industries Co.-0012-Oct 13, 2011
 Goldstream Fuel-0013-Oct 15, 2011
 Red Valley Mining-0014-Oct 16, 2011

Note 1

Use a white font color for the start and the end tags of the variables, (e.g. `<0>` and `</0>`) so that the visual appearance and the layout of the report will not be affected.

Note 2

Before going to production, it is advisable to practice the use of variables on few sample reports. This is to avoid any unpleasant situation of sending wrong data to clients or customers.

Excel User-Defined Variables

While the concept and the usage for the user defined variables are the same with the PDF reports, Excel user defined variables are defined like the following example

| D | E | |
|-----------------------|---|-------------------|
| (*)burstTokens | userVariables | configFile |
| Germany | <0>Alfreds Futterkiste</0>, <1>Berlin</1> | |
| USA | <0>John Steel</0>, <1>San Francisco</1> | |
| | | |

- If required, Excel user-defined variables can be declared in the *userVariables* column from *burst* meta-data sheet
- Similarly with the PDF variables, Excel user-defined variables are enclosed in between <0>value</0>,<1>value</1> and so on.
- The value for *var0* user variable (associated with *Germany* burst token) is *Alfreds Futterkiste* and the value for *var1* is *Berlin*. The values for the *USA* burst token are the ones which are presented in the screenshot.

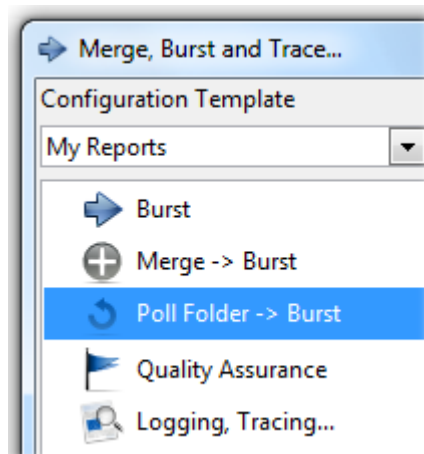
Chapter 6. Automatic Polling for Incoming Reports

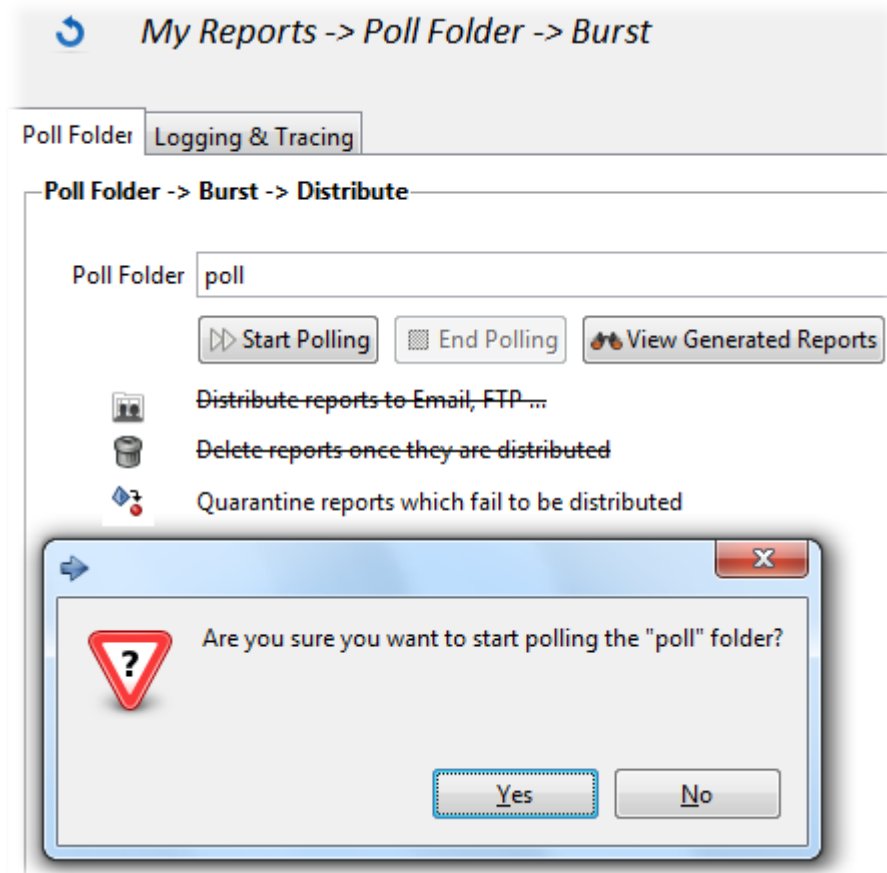
Watch a Folder for Incoming Reports

DocumentBurster can poll a configurable folder and automatically pick for processing all the reports which are dropped to this folder.

This functionality is useful when integrating with an external legacy system which can be configured to generate the reports to the polled folder, thus obtaining an automated system to generate, burst and distribute the reports.

In *DocumentBurster* main window select Actions , Merge, Burst and Trace... , Poll Folder -> Burst





- *Start Polling* - Start watching the selected folder for incoming reports. Once the polling is started, *DocumentBuster* will automatically process all the reports which are dropped to this folder.
- *End Polling* - Stop watching/polling the folder for incoming reports.
- *View Generated Reports* - Browse the generated reports.

Note

The polling GUI presented in this chapter is not available in *DocumentBuster Server*. *DocumentBuster Server* will automatically start to check (poll) the `server/poll` folder once it is started.

For more details about the added features of the *DocumentBuster Server* software please read Chapter 9, *DocumentBuster Server* .

Chapter 7. Quality Assurance

Test for Correctness

When dealing with reports and financial documents it is important to have a good quality assurance mechanism.

I'm concerned with sending misdirected mass email.

This is a legitimate concern which people might have before deciding to implement an automated report delivery system. *DocumentBuster's* quality assurance mechanism is the tool intended to address exactly this concern, thus greatly reducing the risk and increasing the confidence of doing mass report distribution.

Test Mode

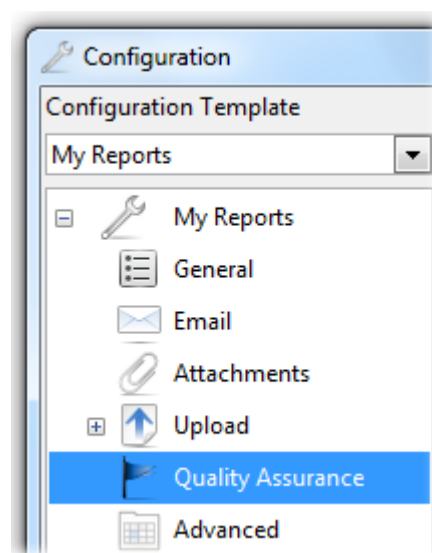
Quality assurance should be used for practicing, before "going live", that everything is configured as it should be. When running a test, all document transmissions can be sent via email to your email address instead of the actual recipients. The emails you receive will include the document(s) as a PDF attachment and the body of the email messages will be the same which the actual recipients are going to receive. Once you are comfortable using *DocumentBuster* you can "go live" and distribute the reports to the actual recipients.

Note

While email distribution is the most common distribution method, *DocumentBuster's* quality assurance mechanism can be similarly used to validate any upload/distribution configuration which the software currently supports (e.g. ftp, file share, etc.)

Configuration

In *DocumentBuster* main window select Actions , Configure , Quality Assurance

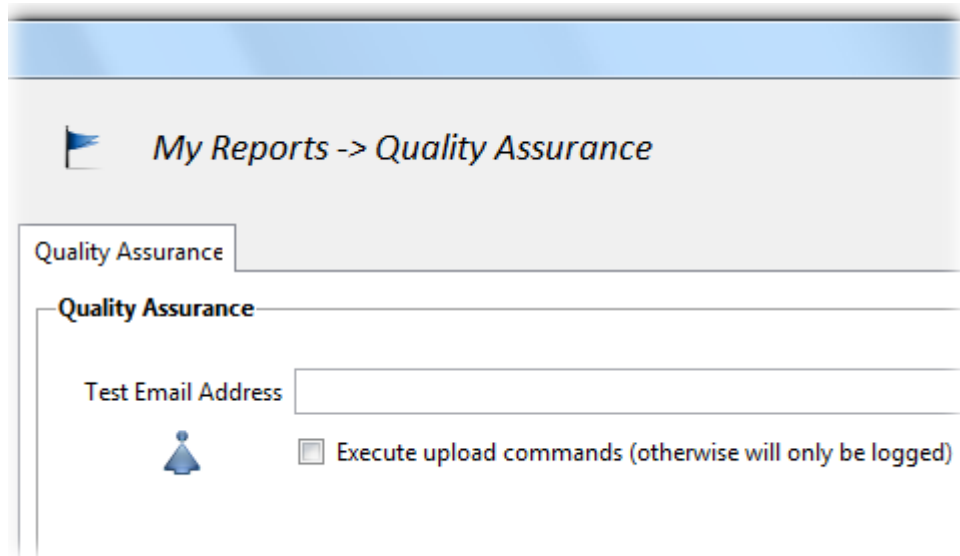


My Reports -> Quality Assurance

Quality Assurance

Quality Assurance

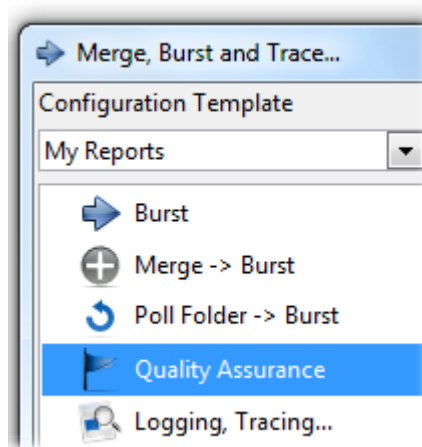
Test Email Address

 ☐ Execute upload commands (otherwise will only be logged)

- *Test Email Address* – Address used to receive the emails when running quality assurance tests.
- *Execute upload commands (otherwise will only be logged)* – If checked, any configured upload command (e.g. ftp, file share, etc.) will be executed when running the test, otherwise upload commands will only be logged. Default value is unchecked.

Run Test

In *DocumentBuster* main window select Actions , Merge, Burst and Trace... , Quality Assurance



One of the following test modes can be selected:

- *Test all the burst tokens which are found in the input report (emails and upload commands will only be logged)* – *DocumentBuster* will test all the burst tokens which are found in the report. If *DocumentBuster* is configured to distribute reports then one *email (or upload) log* file will be generated for each different recipient (burst token). The email/upload log files will contain all the relevant information which is required to validate that the software is configured properly.
- *Test the following burst tokens* – *DocumentBuster* will test the (comma separated) list of burst tokens which are specified by the user. If *DocumentBuster* is configured to distribute reports then the corresponding emails will be sent to the test email address and will also be logged to separate log files. If configured as such then *DocumentBuster* will execute any relevant upload command(s) and will log the commands in the appropriate files. The email/upload log files will contain all the relevant information which is required to validate that the software is configured properly.
- *Test 2(configurable) random burst tokens* – *DocumentBuster* will test 2(configurable) random burst tokens. If *DocumentBuster* is configured to distribute reports then the corresponding emails will be sent to the test email address and will also be logged to separate log files. If configured as such then *DocumentBuster* will execute any relevant upload command(s) and will log the commands in the appropriate files. The email/upload log files will contain all the relevant information which is required to validate that the software is configured properly.

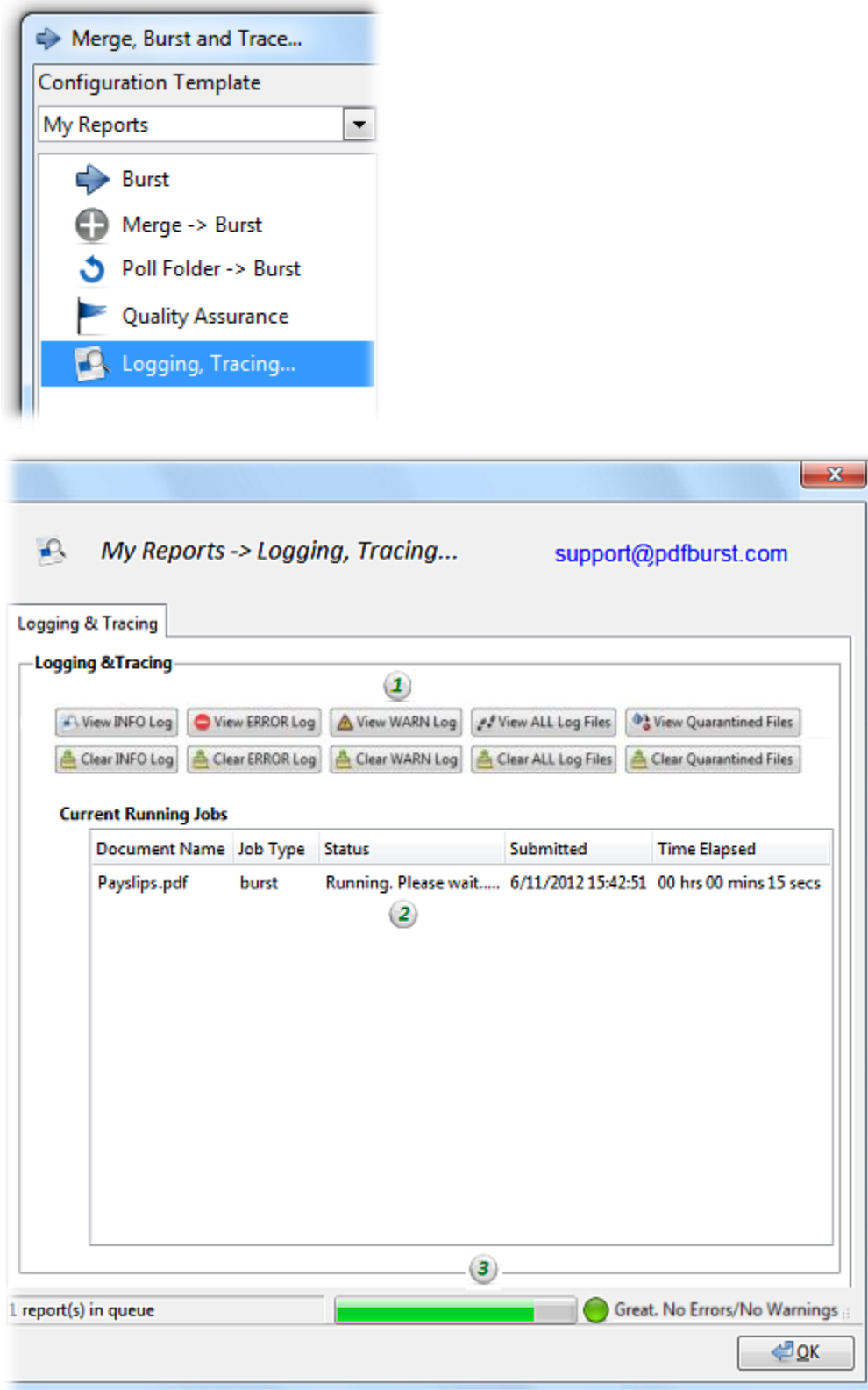
Once the test is finished the output and log files will be available for review by pressing the *View Generated Reports* button.

Logging

It is crucial for the software to properly distribute all the documents to the correct destinations. However, sometimes because of various reasons the distribution of the documents might fail. This might happen because the email server connection details are not correct or because the server itself is down or maybe the SSL settings are not accurate.

DocumentBuster has support for logging all activities and for tracing back the reports which fail to be distributed.

In *DocumentBurst* main window select Actions , Merge, Burst and Trace... , Logging, Tracing...



- (1) - *View/Clear Log and Quarantine Files* - Buttons to view/clear INFO, ERROR or WARN log files. View/Clear Quarantine Files are the buttons to view/clear the reports which are quarantined.
- (2) - *Current Running Jobs* - List with the jobs which are currently executing
- (3) - *Progress and Status Bars* - At the bottom of the screen there are a progress bar which shows the running progress and a status bar which displays possible problems. Green color means that there are no failed reports while the red color means that at least one report failed to be distributed and was quarantined.

By default, in order to keep the log files simple and clear, the program is logging the errors and the minimum necessary number of informative events.

If required, *DocumentBuster* can be configured to generate detailed log files. To do this please edit the file `log4j.xml` and make the following change

```
<!-- Log level value="debug" will make DocumentBuster more verbose -->
<level value="info" />
```

Do the bold change

```
<!-- Log level value="debug" will make DocumentBuster more verbose -->
<level value="debug" />
```

Save the configuration file and run the program again. Now *DocumentBuster* will generate detailed log files which can be used for tracing of possible problems.

Send Processing Errors by Email

DocumentBuster can be configured to send an email whenever a problem is happening. To do this, please edit the file `log4j.xml` and un-comment the following line

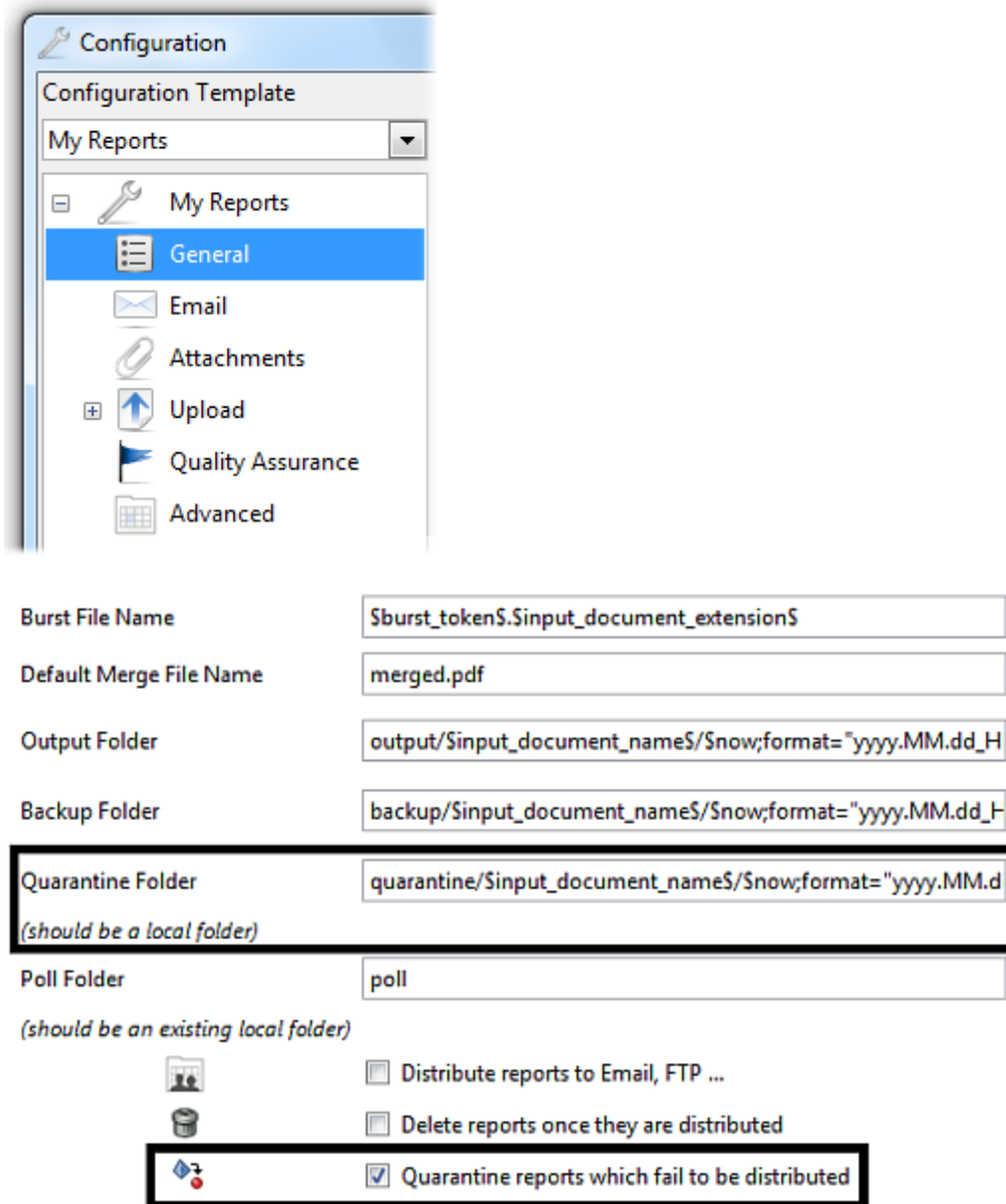
```
<!-- <appender-ref ref="mailAppender" /> -->
```

In addition, it is required to give all the correct email connection details in the *mailAppender* section.

Quarantine Reports Which Fail To Be Delivered

DocumentBuster can be configured to quarantine (save) the documents which fail to be delivered to a configurable folder. The failed documents should be reviewed at a later point in time in order to take a decision (either to distribute again or to do something else).

In *DocumentBuster* main window select Actions , Configure , General



Configuration

Configuration Template
My Reports

My Reports

General

Email

Attachments

Upload

Quality Assurance

Advanced

Burst File Name:

Default Merge File Name:

Output Folder:

Backup Folder:

Quarantine Folder:
(should be a local folder)

Poll Folder:
(should be an existing local folder)

☐ Distribute reports to Email, FTP ...

☐ Delete reports once they are distributed

☒ Quarantine reports which fail to be distributed

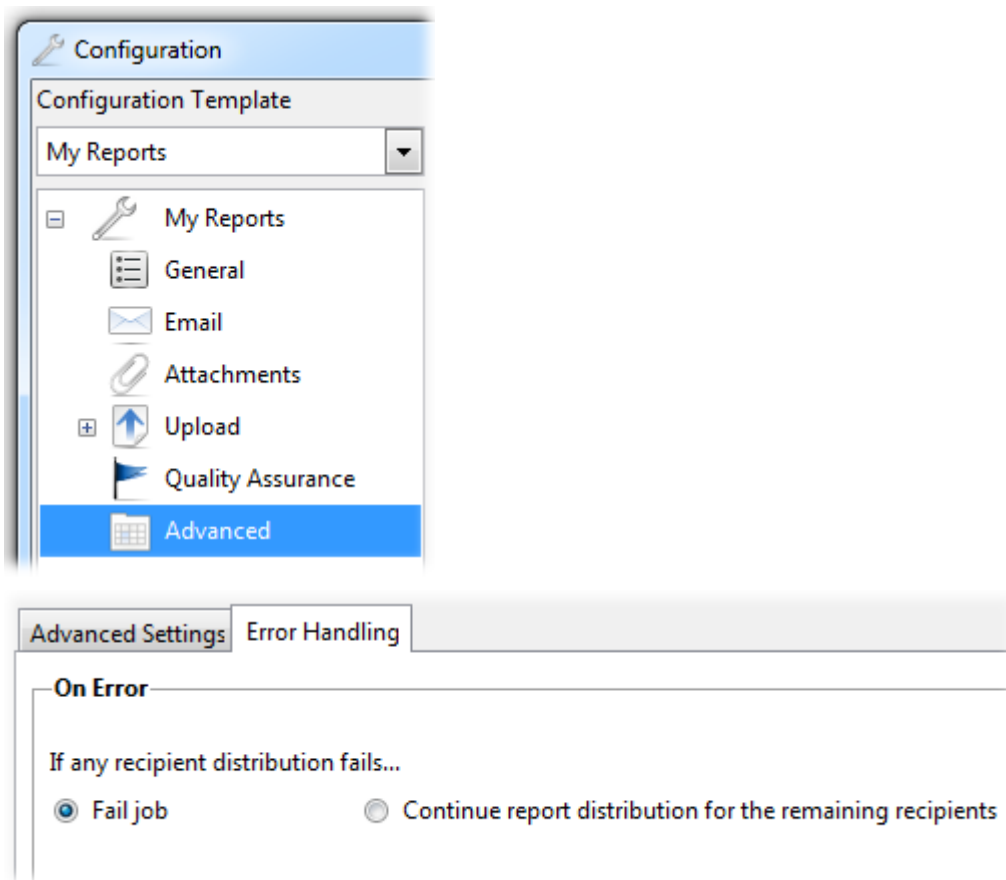
- *Quarantine Folder* – Used to specify the folder where to quarantine the files which fail to be distributed. It should be a local folder and the default value is `quarantine/$input_document_name$/Snow;format="yyyy.MM.dd_HH.mm.ss"$` .
- *Quarantine reports which fail to be distributed* - If checked, the reports which fail to be distributed will be saved to the quarantine folder, otherwise no.

Error Handling

By default *DocumentBuster* software is stopping the report distribution when an error is happening. *DocumentBuster* software will wait for the root cause of the problem to be fixed before triggering again the same report. The reason for this default behavior is that usually the problems are happening because of incorrect or inconsistent data and that all of these problems should be promptly fixed otherwise, when ignoring incorrect data or misconfiguration, bigger problems will probably happen later.

Having said that, there are few legitimate situations in which the continuation of the report distribution might be the preferred choice. For example, you might be looking at distributing reports to 5000 customers out of which 3 customers have an incorrect email address in the database. Maybe the customer data was collected during a long period of time (and by many people) so few data problems are expected. In such a case you will not be willing to halt all the distribution and instead you will need to distribute the reports to all the correct addresses and, in the same time, to get a list with the incorrect email addresses which needs to be fixed afterwards.

In *DocumentBurst* main window select Actions , Configure , Advanced , Error Handling



If any recipient distribution fails...

- *Fail job* - The report distribution is stopped when an error is happening. The offending error is reported through the log file. This is the default configuration.
- *Continue report distribution for the remaining recipients* - If any recipient distribution fails then the distribution continues for the remaining recipients. The offending errors are reported through the log file.

Chapter 8. Command Line

DocumentBurst has a command line interface and can be integrated and executed from existing legacy software systems. All the features of the program are available through command line.

Note

Before running *DocumentBurst* in command line, the software should be properly configured.

Usage

Following is the *DocumentBurst* command line usage

```
usage: DocumentBurst
-b,--burst           Burst the merged file
-c,--configuration <arg> Configuration file
-f,--file <arg>      Input file
-m,--merge <arg>     List of files to merge
-o,--output <arg>     Name of the output merged file
-p,--poll <arg>       Start polling a folder for reports
-ta,--testall         Test all the burst tokens found in the input
                      document
-tl,--testlist <arg>  Comma separated list of burst tokens which
                      should be tested
-tr,--testrandom <arg> Number of random burst tokens which should be
                      tested
```

Burst Reports

Windows

Following is the syntax for running the program

```
documentburst.bat -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>]
```

- -f <pathOfTheFileToBurst> - Mandatory argument. Path to the PDF or Excel report to burst.
- -c <pathOfTheConfigurationFileToUse> - Optional argument. Path to the configuration file to be used.

For example the command

```
documentburst.bat -f samples/Payslips.pdf
```

will burst the *Payslips.pdf* file located in the *samples* folder using the default *config/burst/settings.xml* configuration file.

Unix/Linux

Following is the syntax for running the shell script

```
./documentburst.sh -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>]
```

For example the command

```
./documentburst.sh -f samples/Payslips.pdf
```

will burst the `Payslips.pdf` file located in the `samples` folder using the default `config/burst/settings.xml` configuration file.

Merge Reports

Windows

Following is the syntax for running the program

```
documentburster.bat -m <"pathOfTheFileToMerge1|...|pathOfTheFileToMergeN"> [-o <mergedFileName>] [-b]
```

- `-m <"pathOfTheFileToMerge1|...|pathOfTheFileToMergeN">` - Mandatory argument. List of the PDF reports to merge (separated by the `|` character).
- `-o <mergedFileName>` - Optional argument. The name of the output merged file. If it is not specified then the `merged.pdf` file name is assumed by default.
- `-b` - Optional argument. Optional switch which specifies that the resulted merged file should be also burst.

For example the command

```
documentburster.bat -m "samples/Invoices-Oct.pdf|samples/Invoices-Nov.pdf|samples/Invoices-Dec.pdf" -o Invoices-1st-Quarter.pdf -b
```

will first concatenate the files `Invoices-Oct.pdf`, `Invoices-Nov.pdf` and `Invoices-Dec.pdf` (`-m`) into a file called `Invoices-1st-Quarter.pdf` (`-o`) and will burst the merged file (`-b`).

Unix/Linux

Following is the syntax for running the program

```
./documentburster.sh -m <"pathOfTheFileToMerge1|...|pathOfTheFileToMergeN"> [-o <mergedFileName>] [-b]
```

Poll Folder for Incoming Reports

Windows

Following is the syntax for running the program

```
documentburster.bat -p <pathOfTheFolderToPoll>
```

For example the command

```
documentburster.bat -p poll
```

will start polling the folder `poll` for incoming reports to process.

Unix/Linux

Following is the syntax for running the shell script

```
./documentburster.sh -p <pathOfTheFolderToPoll>
```

For example the command

```
./documentburster.sh -p poll
```

will start polling the folder `poll` for incoming reports to process.

Quality Assurance

Windows

1. Following is the syntax for testing all the burst tokens

```
documentburster.bat -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -ta
```

For example the command

```
documentburster.bat -f samples/Payslips.pdf -ta
```

will test all the burst tokens which are found in the `Payslips.pdf` file located in the `samples` folder using the default `config/burst/settings.xml` configuration file.

2. Following is the syntax for testing a comma separated list of tokens

```
documentburster.bat -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -tl  
<"token1,token2,...tokenN">
```

For example the command

```
documentburster.bat -f samples/Payslips.pdf -tl  
"clyde.grew@northridgehealth.org,alfreda.waldback@northridgehealth.org"
```

will test the tokens `clyde.grew@northridgehealth.org` and `alfreda.waldback@northridgehealth.org` which are found in the `Payslips.pdf` file located in the `samples` folder using the default `config/burst/settings.xml` configuration file.

3. Following is the syntax for testing random tokens

```
documentburster.bat -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -tr <N>
```

For example the command

```
documentburster.bat -f samples/Payslips.pdf -tr 2
```

will test two random tokens which are found in the `Payslips.pdf` file located in the `samples` folder using the default `config/burst/settings.xml` configuration file.

Unix/Linux

1. Following is the syntax for testing all the burst tokens

```
./documentburster.sh -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -ta
```

2. Following is the syntax for testing a comma separated list of tokens

```
./documentburster.sh -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -tl  
<"token1,token2,...tokenN">
```

3. Following is the syntax for testing random tokens

`./documentburster.sh -f <pathOfTheFileToBurst> [-c <pathOfTheConfigurationFileToUse>] -tr <N>`

Part III. *DocumentBuster Server*

Part III presents how to configure and run *DocumentBuster Server* in order to benefit from additional and powerful capabilities like scheduled report distribution, capability to run as windows services and a browser based web interface which can be accessed simultaneously by multiple people.

Chapter 9. *DocumentBurster Server*

DocumentBurster Server can be deployed as a central system to provide report bursting and report distribution services to multiple people or software applications from your organization.

DocumentBurster Server has all the capabilities of *DocumentBurster* and some additional and advanced features like

- **Server Architecture** in order to support multiple people or applications from within your organization
- **Web Based User Interface** which is compatible with all the major browsers - Internet Explorer, Firefox, Chrome, Opera and Safari
- **Windows Services** support in order to achieve unattended report bursting and report distribution
- **Job Monitoring Console** to see the currently executing jobs and trace the status for previously submitted jobs
- **Scheduling** support which allows to define nightly, monthly or custom schedules for the report bursting and report distribution jobs

In addition, if required, *DocumentBurster Server* is a powerful platform which has the foundation to add and tailor other additional report distribution features which an organization might need, for example

- Burst, split and merge any report format (out of the box *DocumentBurster* can process PDF and Excel reports)
- Full indexing and search capabilities for the reports which are being burst and distributed. In simple words this feature allows to quickly find a report which was distributed one year ago - even if your company is distributing a big number of reports each month.

Installation

Prerequisites

DocumentBurster Server software has the same prerequisites like *DocumentBurster*. If required, please read the *DocumentBurster in 5 Minutes* document to install the prerequisites for running the software.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

Download *DocumentBurster Server*

For evaluation purposes, *DocumentBurster Server* can be downloaded from this link - <http://www.pdfburst.com/documentburster-server.zip>.

Extract the zip archive to a drive like C : /

Once the zip file is extracted, a new directory will be available (e.g. - C : /DocumentBurster-5.2.8) which will contain the following two sub-directories

- `server` - contains the binaries and the scripts for starting and stopping the report bursting server
- `web-console` - binaries and scripts for the *DocumentBurster Web Console*

Start and Stop

A. Server

Configuration

DocumentBuster Server is configured using the same GUI interface (*DocumentBuster.exe*) like *DocumentBuster*. *DocumentBuster* configuration was described in detail in the section called “Configuration” .

Starting and Stopping

- Starting - once configured, the server can be started using `server/startServer.bat` (Windows) or `server/startServer.sh` (Linux) scripts.
- Stopping - `server/shutServer.bat` (Windows) or `server/shutServer.sh` (Linux) scripts should be used to stop the server.

Automatic Polling

Once started, the server is automatically checking for new reports to process in the `server/poll` directory. Any report which is dropped to `server/poll` folder is automatically picked up and processed by *DocumentBuster Server*.

B. Web Console

Dependency

Web Console depends on *Server* - When starting, the *web console* is connecting to the *server* so, it is mandatory to have the *server* component started first.

Starting and Stopping

- Starting - once the *server* is started, the *web console* can be started using `web-console/startConsole.bat` (Windows) or `web-console/startConsole.sh` (Linux) scripts.
- Stopping - `web-console/shutConsole.bat` (Windows) or `web-console/shutConsole.sh` (Linux) scripts should be used to stop the console.

Once started, *DocumentBuster Web Console* application can be accessed by typing following URL in your browser

`http://machine-name:8080/burst`

for example

`http://localhost:8080/burst`

Note

Sometimes, due to wrong configuration, the web console might fail to start properly. The solution to this problem can be found by following the section called “ Windows - *DocumentBuster Web Console* Is Failing to Start? ” (troubleshooting entry).

Web Console

DocumentBuster Server is coming with a web based interface which can be accessed from any major web browser (e.g. Mozilla Firefox, Internet Explorer, Google Chrome, etc.). The web interface can be used for triggering new ad hoc jobs, scheduling jobs for later execution or for viewing the status, history and detailed logs of the previously submitted jobs.

The web console needs the server to be started, so make sure it is so. After the server is started, in the folder where the software was extracted, please execute `web-console/startConsole` script to get the console started. After few seconds the application can be accessed by typing following URL in the browser

`http://machine-name:8080/burst`

for example

`http://localhost:8080/burst`

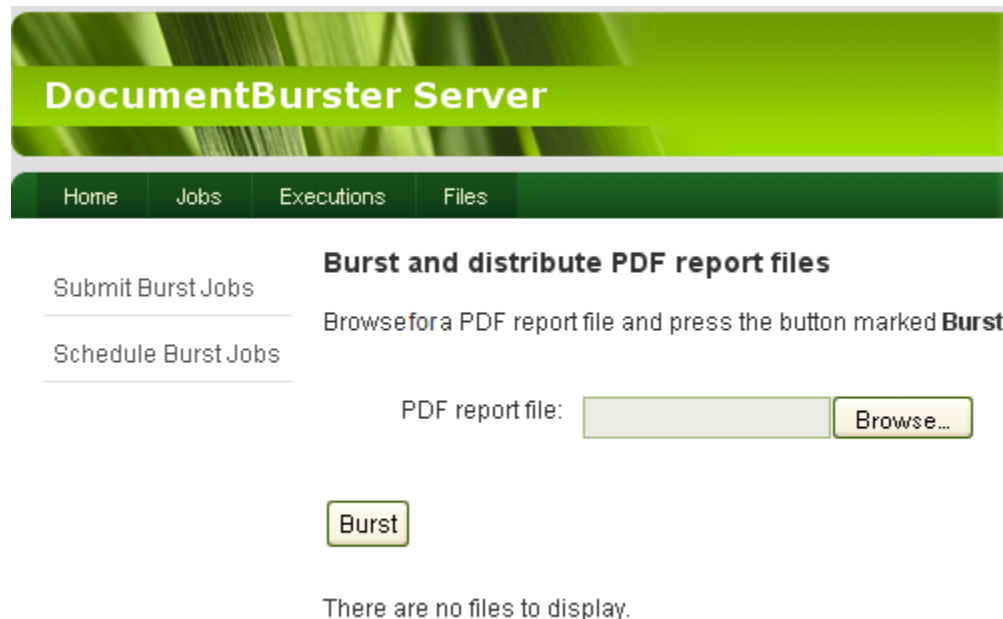
Ad hoc/Immediate Jobs - Reports can be immediately burst and distributed through the *Files - Submit Burst Jobs* menu entry.

Scheduled Jobs can be submitted through the *Files - Schedule Burst Jobs* menu entry. Uploaded report files are placed in the `server/input-files/scheduled` folder being scheduled for execution at a later time.

DocumentBuster Web Console can be used to submit new jobs for immediate execution, to schedule jobs for later execution, to view the currently running jobs or to check the status, history and the logs of the previously submitted jobs.

Screenshots

Following are some screenshots from the application.



The screenshot shows the DocumentBuster Server web console. At the top is a green header with the text "DocumentBuster Server". Below the header is a navigation bar with tabs: "Home", "Jobs", "Executions", and "Files". The "Files" tab is currently selected. On the left side, there are two links: "Submit Burst Jobs" and "Schedule Burst Jobs". The main content area is titled "Burst and distribute PDF report files". It contains the instruction: "Browse for a PDF report file and press the button marked **Burst**." Below this, there is a form with a label "PDF report file:" followed by a text input field and a "Browse..." button. Below the input field is a "Burst" button. At the bottom, it says "There are no files to display."

Burst reports - uploaded reports are picked up and processed by the server

The screenshot shows the DocumentBuster Server web interface. At the top is a green header with the text 'DocumentBuster Server'. Below the header is a navigation bar with links: Home, Jobs, Executions, and Files. The main content area has a left sidebar with 'Submit Burst Jobs' and 'Schedule Burst Jobs' links. The 'Schedule Burst Jobs' link is active. The main content area is titled 'Schedule report bursting and report distribution jobs'. Below the title, it says 'Browse for a PDF report file and press the button marked **Schedule**'. There is a text input field for 'PDF report file:' followed by a 'Browse...' button. Below this is a 'Schedule' button. At the bottom, it says 'There are no files to display.'

Schedule report bursting jobs - schedule documents for distribution at a later time

The screenshot shows the DocumentBuster Server web interface. At the top is a green header with the text 'DocumentBuster Server'. Below the header is a navigation bar with links: Home, Jobs, Executions, and Files. The main content area has a left sidebar with 'Job Names Registered' link. The main content area is titled 'Job Names Registered'. Below the title is a table with the following data:

| Name | Description | Execution Count | Launchable | Incrementable |
|-----------------|----------------|-----------------|------------|---------------|
| <u>burstJob</u> | No description | 2 | false | false |

Rows: 1-1 of 1 Page Size: 20

Jobs page

DocumentBurster Server

[Home](#) | [Jobs](#) | [Executions](#) | [Files](#)

Recent and Current Job Executions

[Stop All](#)

| ID | Instance | Name | Date | Start | Duration | Status | ExitCode |
|----------|----------|----------|------------|----------|----------|-----------|-----------|
| <u>1</u> | 1 | burstJob | 2011-01-28 | 19:44:10 | 00:00:00 | COMPLETED | COMPLETED |
| <u>0</u> | 0 | burstJob | 2011-01-28 | 19:28:18 | 00:00:40 | FAILED | FAILED |

Rows: 1-2 of 2 Page Size: 20

DocumentBurster Server job executions page - view status and history of distribution jobs

Details for Step Execution

| Property | Value |
|---------------|--|
| ID | 0 |
| Job Execution | <u>0</u> |
| Job Name | burstJob |
| Step Name | burstStep |
| Start Date | 2011-01-28 |
| Start Time | 17:28:18 |
| Duration | 00:00:40 |
| Status | FAILED 1 |
| Exit Code | FAILED 2 |
| Exit Message | <div> <div>2</div> <div> <p>java.lang.Exception: Free version limit - Free DocumentBurster can burst</p> <p>http://www.pdfburst.com/ at com.smartwish.documentburster.engine.Pd</p> <p>sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.ref</p> <p>Source) at java.lang.reflect.Method.invoke(Unknown Source) at</p> <p>org.codehaus.groovy.runtime.callsite.PojoMetaMethodSite\$PojoCachedM</p> <p>org.codehaus.groovy.runtime.callsite.PojoMetaMethodSite.call(PojoMeta</p> <p>org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallS</p> <p>burst.run(burst.groovy:20) at groovy.util.GroovyScriptEngine.run(GroovyS</p> <p>com.smartwish.batch.ext.Scriptlet.execute(Scriptlet.java:36) at com.sma</p> <p>org.springframework.batch.core.step.tasklet.Tasklet\$Step\$ChunkTransp</p> </div> </div> |

Detailed information for the selected job

- (1) - Status of the job
- (2) - Logging details for the failed jobs

Run *DocumentBuster* at System Startup

Being a server application, *DocumentBuster* can be configured to run in the background as long as the operating system is running.

Following paragraphs will detail how to configure *DocumentBuster* to run automatically when the system is starting. The screenshots are taken from Windows 7 and the same can be similarly achieved on any other Windows distribution.

DocumentBuster software can start when the Microsoft Windows operating systems is booted, either

- By configuring *DocumentBuster* to run as standard **Windows services**, or
- By **scheduling** *DocumentBuster* to run *When the computer starts*

The above options are two different ways for solving similar requirements. You are free to choose the option which is best fitting your needs.

Following is how to configure each option in detail.

Windows Services

Both the *server* and the *web console* components can be separately configured to be executed as Windows services.

The *server* is the mandatory component which is doing the report processing while the *web console* will be started as a web GUI interface for posting the reports to the server.

DocumentBuster Web Console depends on the server to be started and, as a result, the *DocumentBuster Web Console* Windows service will be automatically configured to depend on the *DocumentBuster Server* Windows service.

A. Configure *DocumentBuster Server* as a Windows service

The server will be configured as a service using the `server/service.bat` script.

Step 1

In the command prompt, using the `cd` command, switch to the folder where the server is installed. For example, assuming *DocumentBuster* is extracted in the folder `C:\DocumentBuster` then



Step 2

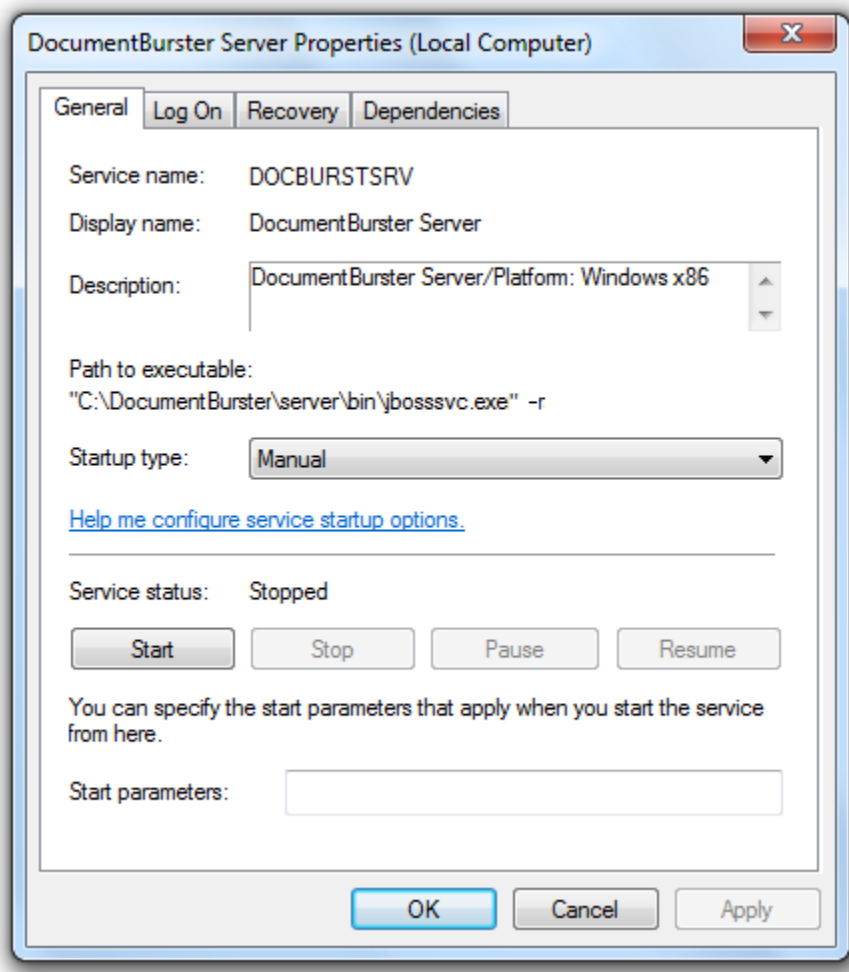
Execute the command *service.bat install*

```
C:\WINDOWS\system32\cmd.exe

C:\DocumentBuster\server>service.bat install
Service DocumentBuster Server installed

C:\DocumentBuster\server>
```

DocumentBuster Server Windows service is now installed and it is properly listed in *Control Panel -> Administrative Tools -> Services*.

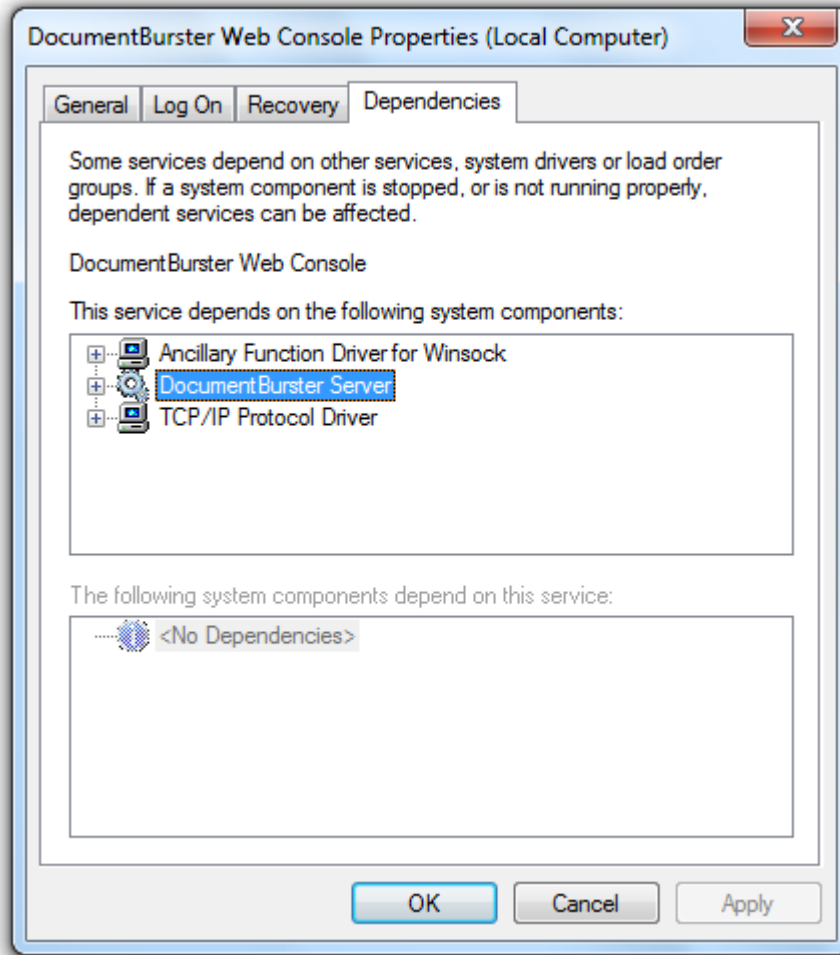


By default the service is stopped and it is configured as having the *Startup Type 'Manual'*.

B. Configure *DocumentBuster Web Console* as a Windows service

DocumentBuster Web Console will be configured as a Windows service by following the same steps which were previously presented for the *DocumentBuster Server* component. The web console will be configured as a service using the `web-console/console/bin/service.bat` script.

DocumentBuster Web Console Windows service will be automatically configured to depend on the previously installed *DocumentBuster Server* Windows service.



Note

On Windows Server 2008 R2 it is required to execute the *service.bat install* DOS commands with elevated or administrative privileges. To run these commands, you can use the *Run as administrator* command

- Click the *Start* button
- In the Search box, type *command prompt*
- In the list of results, right-click *Command Prompt* , and then click *Run as administrator* . If you are prompted for an administrator password or confirmation, type the password or provide confirmation.

Note

If required, both *DocumentBuster Server* and *DocumentBuster Web Console* Windows services can be individually uninstalled from the system by executing the command *service.bat remove* on their corresponding script file.

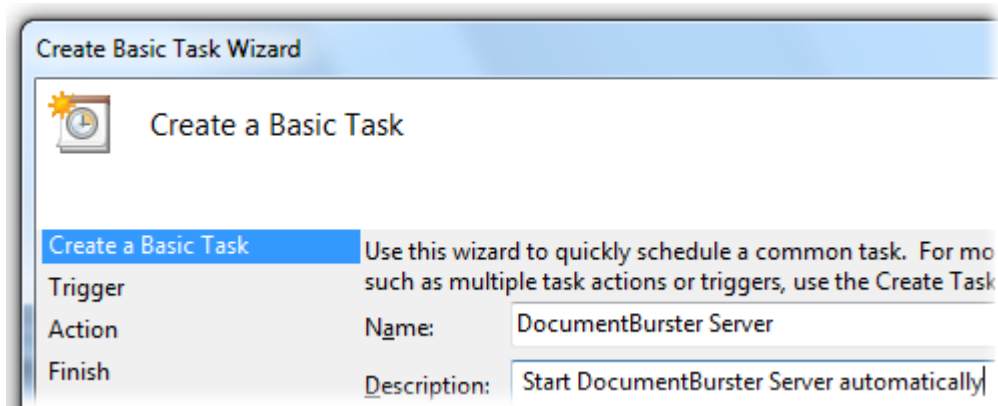
Before uninstalling, please make sure that the services are properly stopped.

Windows Scheduling

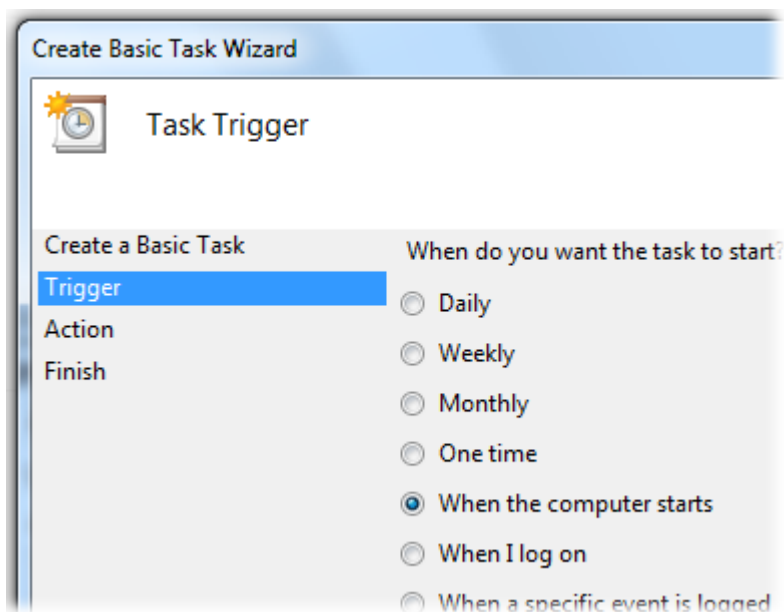
If you find too heavy running *DocumentBuster* as Windows Services then Windows Scheduled Tasks is another option for getting *DocumentBuster* to automatically start when Windows is starting.

The screenshots are showing how to schedule `server/startServer.bat` script in order to automatically start the *server* component.

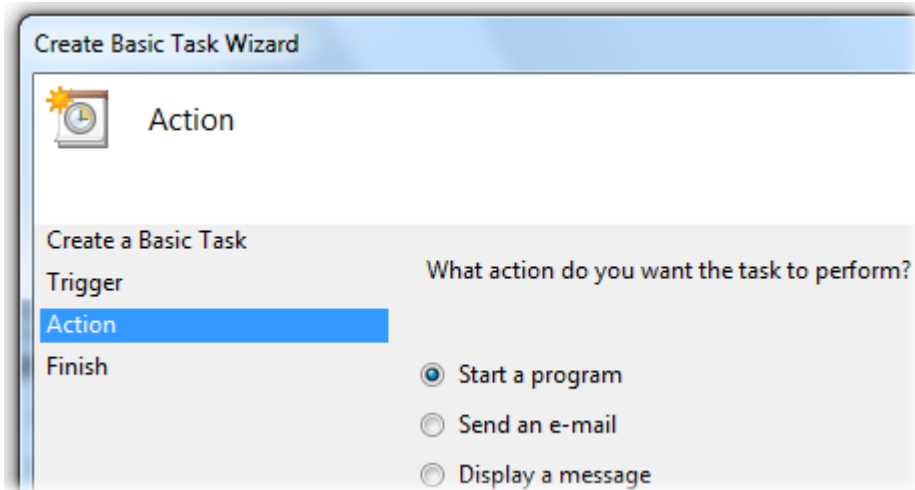
1. In *Windows 7* select Start , Search , Task Scheduler , Create Basic Task...



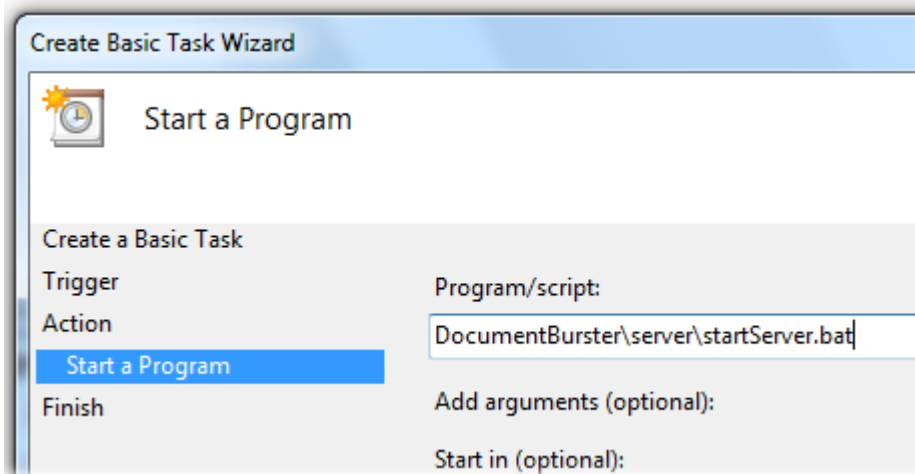
2. Click *Next*



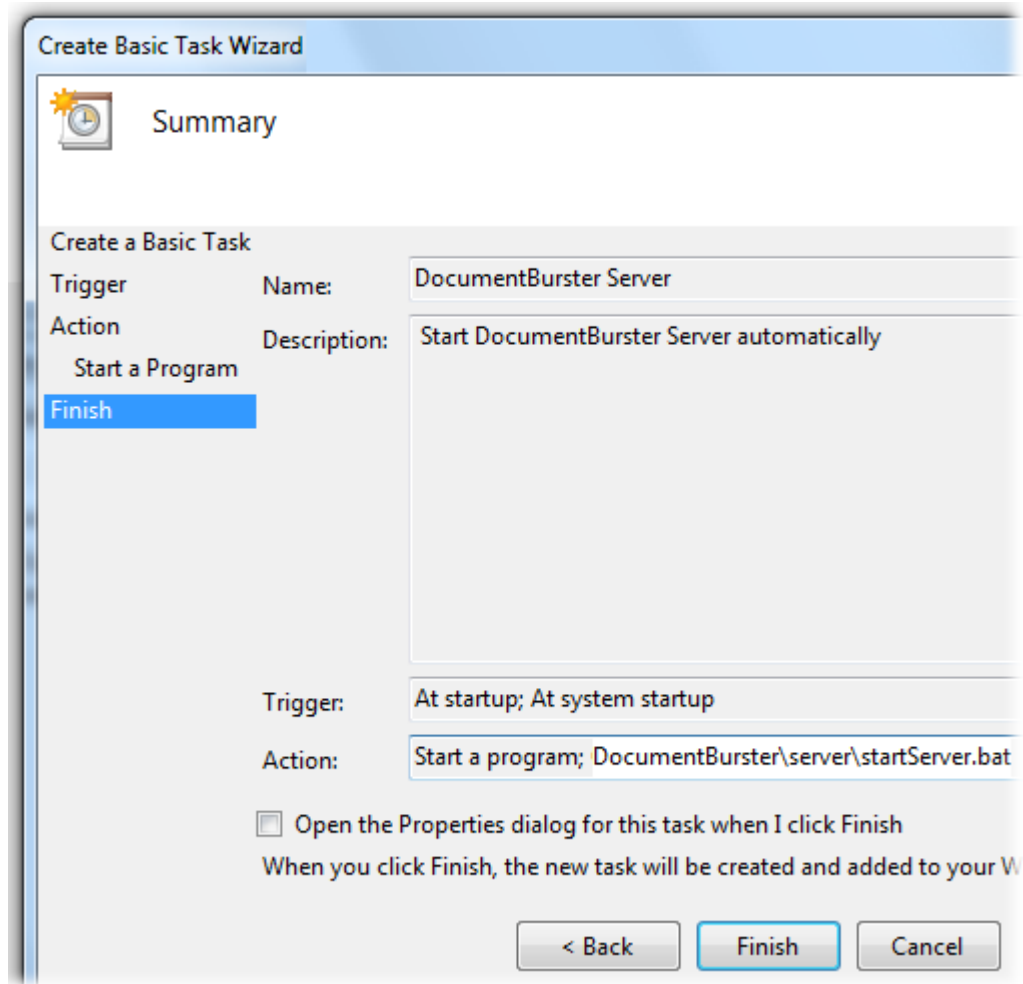
3. Click *Next*



4. Click *Next*



5. Click *Next* and then *Finish* to get the task scheduled.



7. Done. `server/startServer.bat` script was scheduled to start when Windows is starting.

Note 1

The above screenshots show how to schedule the `server/startServer.bat` script in order to automatically start the *server* component. The same can be done for `web-console/startConsole.bat` script in order to automatically start the *DocumentBuster Web Console*.

Note 2

DocumentBuster Web Console depends on the server to be started and, as a result, the `web-console/startConsole.bat` batch file should be scheduled to start just after the `server/startServer.bat` component had been started.

Scheduling

DocumentBuster Server can handle scheduled report bursting and report distribution jobs. By default the software can handle jobs scheduled for nightly (midnight) execution. If this is what it is required, then it is nothing more to be configured in regards with scheduling. On the other hand, it is possible to customize

the scheduling - familiarity with other cron like schedulers will help in understanding better the scheduling mechanism implemented in *DocumentBuster*. Yearly, monthly, weekly, daily, hourly or any other custom report bursting schedule jobs are all possible to define.

While for executing ad hoc, immediate report bursting jobs, *DocumentBuster Server* is checking the `server/poll` folder, scheduled reports should be placed in the `server/input-files/scheduled` directory. *DocumentBuster* will properly trigger the report bursting and report distribution jobs to happen at the correct date and time, depending on how the scheduling is configured.

Configuration

DocumentBuster Server scheduling is configured using *cron expressions*.

By default the scheduler is configured to run daily, at midnight. In order to change the default (daily) configuration you need to update the cron expression entry at the end of the configuration file

```
server/config/batch/internal/batch-context.xml
```

See below sample of the cron (scheduling) entry

```
<task:scheduled-tasks scheduler="scheduler">
  <task:scheduled ref="scheduled" method="run" cron="0 0 0 * * ?" />
</task:scheduled-tasks>
```

The text `cron="0 0 0 * * ?"` is the *cron expression* of particular interest in regards with configuring the scheduling. `0 0 0 * * ?` is the encoding to configure the default daily (midnight) schedule. The previous default cron expression can be replaced with any other valid expression, based on the requirements, in order to schedule yearly, monthly, weekly, daily - at different time or hourly report processing jobs. *Cron expression* documentation is out of the scope of this user guide, more details about how to configure a cron expression can be found at

- *CRON expression* - *Wikipedia* - http://en.wikipedia.org/wiki/CRON_expression
- *CRON expression* - *Quartz documentation* - <http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html>

Note

A *DocumentBuster Server* restart is required whenever the cron expression configuration is changed.

Part IV. Appendices

Appendix A, *How to Do This and That*
Appendix B, *SharePoint Report Delivery*
Appendix C, *DocumentBuster Performance*
Appendix D, *Troubleshooting*

Appendix A. How to Do This and That

How To: Distribute External Reports?

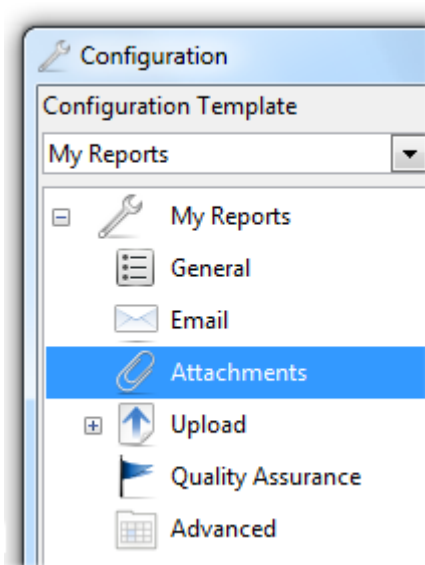
Sometimes it might be required to distribute existing reports which are pre-generated and already contain (only) the relevant data for the intended recipient. Such reports don't require any further splitting but only to be distributed, without any modification, to the proper destination. This destination might be email or any other destination type which is currently supported by *DocumentBuster* (e.g. FTP, SFTP, etc.).

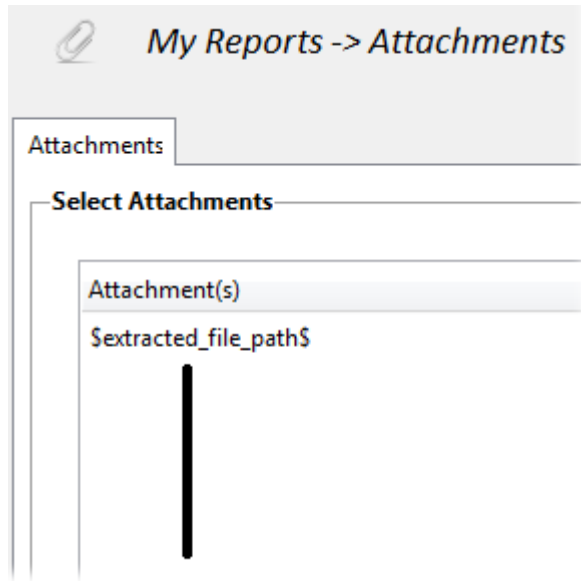
DocumentBuster can be configured to distribute external/existing documents which do not require splitting.

Note

You need to make sure that the reports contain (only) the relevant data for the corresponding recipients.

Configuration





My Reports -> Attachments

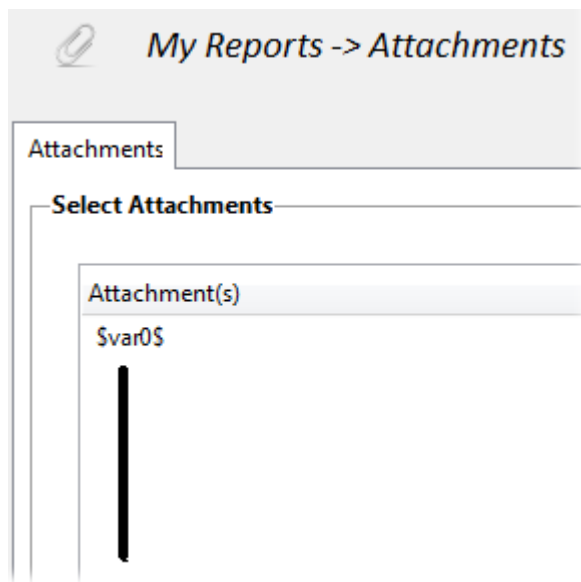
Attachments

Select Attachments

Attachment(s)

\$extracted_file_path\$

\$extracted_file_path\$ is the default configuration. This should be changed with a dynamic value to be filled at run-time (e.g. *\$var0\$* user variable).



My Reports -> Attachments

Attachments

Select Attachments

Attachment(s)

\$var0\$

\$var0\$ should be configured to point to the appropriate document path to be distributed for each recipient. The external documents can be in any format, including PDF, Excel, Word, CSV, Txt, Xml or any other report format.

Recipients List and Document(s) Paths

There are two different ways to configure the recipients list and the path to the document(s) to be distributed to each separate recipient.

- Provide the recipients and the document(s) paths in an Excel sheet
- Provide the recipients and the document(s) paths in a PDF report

Note

These are two different ways for achieving the same business requirements. The alternative which is best fitting the needs should be used.

Provide the recipients and the document(s) paths in an Excel sheet

When using Excel for configuration

- Recipients list can be provided using the *(*)burstTokens* column from the metadata ***burst*** sheet
- The paths to the document(s) to be distributed can be provided using user variables, more specifically using the *userVariables* column from the metadata ***burst*** sheet. For example, mapping *\$var0\$* user variable to the path of the document to be distributed, it is possible to distribute a separate report to each distinct recipient/burst token.

Check the existing sample report `samples/Distribute-External-Files.xls` which can be found in `samples` folder. `Distribute-External-Files.xls` report contains a list of recipients together with the relevant report to be distributed to each one of them.

| D | E |
|--|---|
| (*)burstTokens | userVariables |
| clyde.grew@northridgehealth.org | <0>samples/Invoices-Oct.pdf</0> |
| kyle.butford@northridgehealth.org | <0>samples/Invoices-Nov.pdf</0> |
| alfreda.waldback@northridgehealth.org | <0>samples/Invoices-Dec.pdf</0> |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| Recipients list is provided using the (*)burstTokens column | The path of the documents is dynamically configured using the \$var0\$ user variable |

The document paths are dynamically mapped using the *\$var0\$* user variable. In this sample

- clyde.grew@northridgehealth.org will receive by email `samples/Invoices-Oct.pdf`,
- kyle.butford@northridgehealth.org will get `samples/Invoices-Nov.pdf` and
- alfreda.waldback@northridgehealth.org will get `samples/Invoices-Dec.pdf`.

Fetch recipients list from other data sources

If required, using Excel, it should be possible to fetch the recipients list and the document(s) paths from other data sources.

Excel has the capability to connect and fetch data from various data sources including SQL Server, Oracle, Microsoft Access, MySQL and most of the existing database types. Using Excel it is also possible to fetch data from existing Xml, CSV, Txt or other Excel reports.

For more details on how to use Excel to connect to an external data source, please consult the official Microsoft Excel documentation.

Provide the recipients list and the document(s) paths in a PDF report

As an alternative to Excel, it is also possible to provide the recipients and the document(s) paths in a PDF report. This might be convenient when there is already in place an automated PDF based reporting system which can generate the recipients list.

The PDF report should contain one distinct page for each intended recipient and each page should contain all the details required for the specific recipient (including the path(s) to the document(s) to be distributed).

Check the existing sample report `samples/Distribute-External-Files.pdf` which can be found in `samples` folder. `Distribute-External-Files.pdf` report has three pages, each page containing the path to the document to be distributed to the corresponding recipient

- `clyde.grew@northridgehealth.org` ,
- `kyle.butford@northridgehealth.org` and
- `alfreda.waldback@northridgehealth.org`

How To: Implement Conditional Report Distribution?

Using *DocumentBuster* it is possible to implement *conditional report delivery* . While bursting the reports, *DocumentBuster* can check and can distribute only those reports which fail to meet a specified user defined condition and can be configured to skip the delivery for the reports which meet the condition.

Use case example

DocumentBuster can be deployed and can be easily configured to act as a *management by exception* reporting system which will alert the management team only for the cases where the minimum performance levels are not met.

For example, the management of your organization might require for getting each morning only the reports for the sales with very low profit margins or for the transactions with deviations from the approved discounts.

Configuration

By default, if configured to distribute reports, *DocumentBuster* will deliver to the relevant recipients all reports which are being generated during the bursting process. The default behavior can be changed, and *DocumentBuster* can be configured to skip the report distribution, by placing a text which has the pattern similar with `<skip>true</skip>` in each of the output documents which should be skipped for report distribution.

Note 1

The same result can be achieved by using the shorter (while being less expressive) pattern `<s>true</s>`.

Note 2

If configured to distribute reports, *DocumentBuster* will distribute to the relevant recipients all the reports which are being generated during the bursting process and for which *DocumentBuster*

- Is not able to find any explicit positive skip instruction like `<skip>true</skip>`(or `<s>true</s>`) which have the meaning to skip the distribution for the current report
- Finds an explicit negative skip instruction like `<skip>false</skip>`(or `<s>false</s>`) which have the meaning to distribute the current report
- Doesn't find any (positive or negative) skip instruction

Note 3

Depending on the requirements, the report writer software should properly fill the appropriate `<skip>true/false</skip>` tags into the reports which needs to be burst and distributed. Usually the value (*true/false*) from within the `<skip>true/false</skip>` instruction tags will be calculated by a formula defined in the report writer engine and having a business meaning such as

- Very low profit margins
- Discounts higher than the maximum approved ones
- Low inventory levels
- Production runs with very high scrap rates
- Any other formula which makes business sense for your organization

skip_current_file_distribution_if.groovy

This sample script can be used to achieve complex *conditional report delivery* scenarios.

DocumentBurst's built-in capabilities (skip instruction) can be used to achieve many conditional distribution scenarios while this sample script (`scripts/burst/samples/skip_current_file_distribution_if.groovy`) should be used for achieving the remaining and more complex situations which cannot be easily implemented using the built-in *skip* instruction approach.

This sample script can be used to achieve *conditional report distribution* in situations similar with the following

- The condition to skip the distribution cannot be achieved using a *skip* report formula (e.g. skip the delivery for files which are bigger than 20MB)
- The condition to skip the distribution is too complex and it might be more convenient to describe this condition in scripting than with a *skip* report formula
- The input report cannot be modified (for whatever reason) to accommodate any *skip* instruction

`skip_current_file_distribution_if.groovy` approach is described in detail in *DocumentBurst - Advanced Report Delivery Scenarios -> Using Scripts to Achieve More -> Sample Scripts*

<http://www.pdfburst.com/advanced-report-delivery.pdf>

Appendix B. SharePoint Report Delivery

DocumentBuster can be used to deliver reports to Microsoft SharePoint portals.

Microsoft SharePoint is a good document management system which has the following document related capabilities

- Store, organize, and locate documents
- Ensure the consistency of documents
- Manage metadata for documents
- Help protect documents from unauthorized access or use
- Ensure consistent business processes (workflows) for how documents are handled

DocumentBuster is usually used together with SharePoint in order to solve the following business situations

- For connecting two important systems of any organization, the reporting system and the document management system (i.e. SharePoint).
- For publishing relevant reports to be accessible by clients, employees, etc. through the web. Think to the situation where you have few hundreds or thousands of customers or dealers and, with a single click, you can make the relevant individual reports to be available to each one of them on the portal.

Following paragraphs will describe how to configure *DocumentBuster* in order to burst and deliver reports to Microsoft SharePoint portal.

Distribute Reports to SharePoint by Email

The simplest way to upload reports to SharePoint is through *DocumentBuster* 's email distribution capability. *DocumentBuster* email configuration is described in the section called “Distribute Reports by Email” .

From *DocumentBuster* 's point of view, email configuration is exactly the same regardless if the email destinations are normal email addresses or are SharePoint enabled email addresses.

Microsoft SharePoint should be properly configured (by a SharePoint administrator) in order to accept *inbound emails*.

SharePoint Configuration

It is out of the scope of this manual to give full details about SharePoint inbound email configuration. The official Microsoft SharePoint documentation together with the many existing SharePoint books are the places to look for complete documentation.

Instead, this manual will give hints and will further refer to the official documentation for the topics which are of interest when it comes to SharePoint and *DocumentBuster* integration.

Following are the main configuration points in regards with SharePoint inbound email

- Install and configure the SMTP service
- Configure incoming e-mail settings
- Configure incoming e-mail on SharePoint sites

The inbound email configuration should be similar for both SharePoint 2007 and SharePoint 2010 and for detailed configuration the official SharePoint documentation should be consulted.

SharePoint 2007 Resources

Following are some help resources for SharePoint 2007 inbound e-mail configuration

- *Plan incoming e-mail (Windows SharePoint Services)* - <http://technet.microsoft.com/en-us/library/cc288433%28office.12%29.aspx>
- *Configure incoming e-mail settings (Windows SharePoint Services)* - <http://technet.microsoft.com/en-us/library/cc287879%28office.12%29.aspx>
- *Enable and configure e-mail support for a list or library* - <http://office.microsoft.com/en-us/sharepoint-server-help/enable-and-configure-e-mail-support-for-a-list-or-library-HA010082307.aspx?pid=CH101237651033>
- *Demo: Configure a SharePoint Server 2007 site to receive e-mail* - <http://office.microsoft.com/en-us/sharepoint-server-help/demo-configure-a-sharepoint-server-2007-site-to-receive-e-mail-HA010204792.aspx?CTT=3>

SharePoint 2010 Resources

Following are few SharePoint 2010 help resources related with inbound email configuration

- *Plan incoming e-mail (SharePoint Foundation 2010)* - <http://technet.microsoft.com/en-us/library/cc288433.aspx>
- *Configure incoming e-mail (SharePoint Foundation 2010)* - <http://technet.microsoft.com/en-us/library/cc287879.aspx>

Sample - Burst `samples/Payslips.pdf` Report to SharePoint 2007

This example is demonstrating how to distribute reports to a WSS3.0 - SharePoint 2007 portal which is installed on Windows Server 2003.

While bursting `samples/Payslips.pdf` report, *DocumentBuster* will generate three output reports

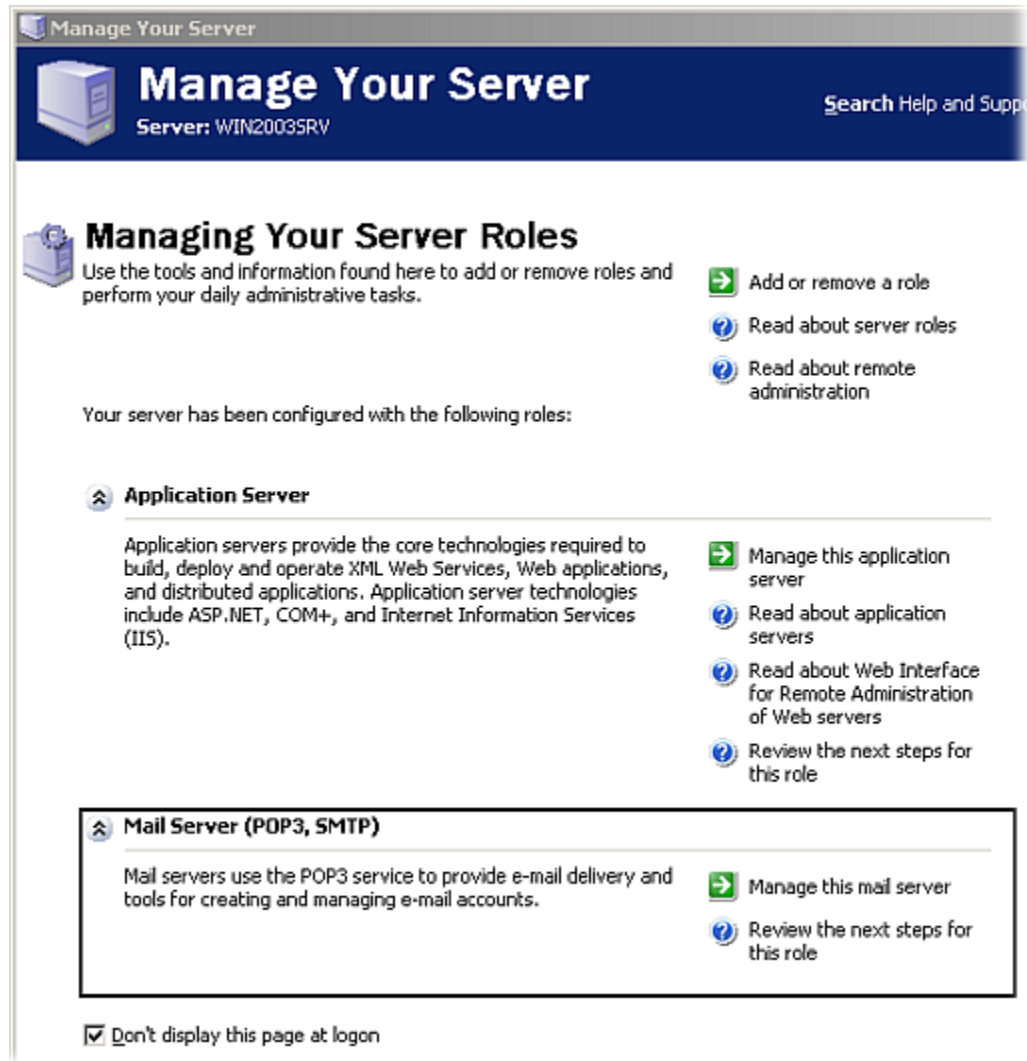
- `alfreda.waldback@northridgehealth.org.pdf`,
- `clyde.grew@northridgehealth.org.pdf`
- `kyle.butford@northridgehealth.org.pdf`

which will be delivered to SharePoint's 2007 default `Shared Documents` document library folder.

Following configuration steps are presented with the help of few screenshots

- Step 1. Control panel - Double check that the SMTP service is properly configured and enabled
- Step 2. SharePoint Central Administration - Configure incoming e-mail settings
- Step 3. SharePoint specific site/library folder - Configure incoming e-mail on SharePoint sites
- Step 4. Configure *DocumentBurst*
- Step 5. Burst samples/Payslips.pdf

Step 1 - Control panel - Double check that the SMTP service is properly configured and enabled



It is OK, SMTP service is available under *Start -> Control Panel -> Administrative Tools -> Manage Your Server*

Step 2 - SharePoint Central Administration - Configure incoming e-mail settings

For this simple demonstration the incoming email is configured in *SharePoint Central Administration* as being *Automatic*. This configuration is done in *Start -> Control Panel -> Administrative Tools -> SharePoint 3.0 Central Administration*

Administration

Central Administration

Operations Application Management

Central Administration > Operations > Incoming E-Mail Settings

Configure Incoming E-Mail Settings

Use this page to change the e-mail settings for this server. You can enable or disable incoming e-mail, [incoming e-mail](#).

Enable Incoming E-Mail

If enabled, SharePoint sites can receive e-mail. If disabled, e-mail and groups will need to be configured individually.

In automatic mode, all required settings are configured automatically. Only if you are not using the SMTP service to deliver e-mail, you need to specify the e-mail drop folder.

Enable sites on this server to receive e-mail?

☒ Yes ☐ No

Settings mode:

☒ Automatic ☐ Advanced

Directory Management Service

The Microsoft SharePoint Directory Management Service connects an organization's user directory in order to provide support for the creation and management of user accounts. The service also creates contacts in your organization's address book for enabled SharePoint lists in their address book.

To use the Directory Management Service you must configure the Central Administration application pool account with the appropriate permissions to the Directory. Alternatively you can configure the Directory Management Web Service.

Use the SharePoint Directory Management Service to manage user accounts.

☒ No ☐ Yes ☐ Use remote

On the bottom of the same screen (SharePoint Central Administration screen) , provide an *E-mail server display address* and select *Accept e-mail from all email servers*

To use the Directory Management Service Administration application pool account with the Directory. Alternatively you can configure the Directory Management Web Service.

Incoming E-Mail Server Display Address

Specify the e-mail server address that will be used for incoming e-mail address for a site, list, or document library.

This setting is often used in conjunction with the Directory Management Web Service to provide a more friendly e-mail address.

Safe E-Mail Servers

Specify whether to restrict the set of e-mail servers that are allowed to send mail to this site. This setting can help ensure the authenticity of incoming e-mail.

E-mail server display address:

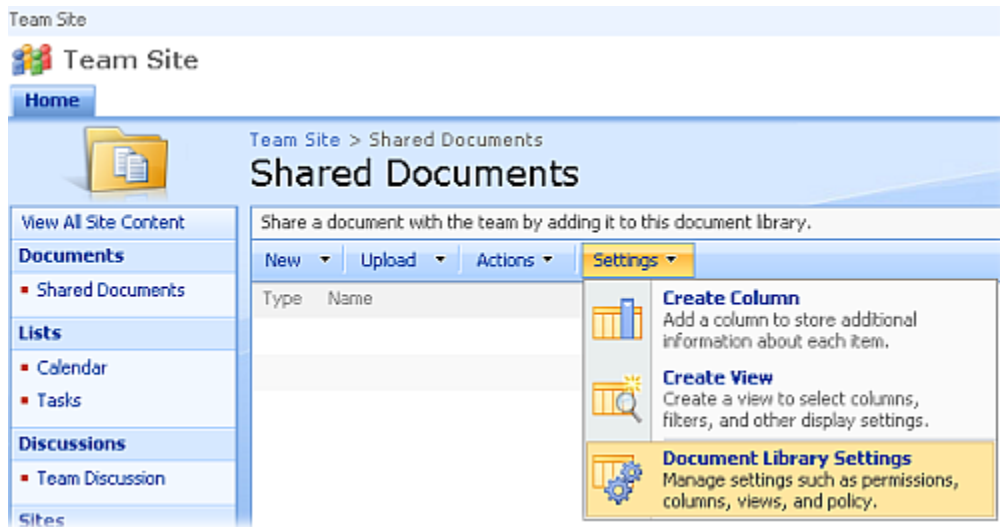
mylist@

For example, mylist@example.com

☒ Accept mail from all e-mail servers

☐ Accept mail from these safe e-mail servers:

Step 3 - SharePoint specific site/library folder - Configure incoming e-mail on SharePoint sites



For the specific document library of interest, enable the incoming e-mail

Team Site > Shared Documents > Settings

Customize Shared Documents

List Information

Name: Shared Documents

Web Address: <http://win2003srv/Shared Documents/Forms/AllItems.aspx>

Description: Share a document with the team by adding it to this document library

E-Mail Address: documentburster@sharepoint

| General Settings | Permissions and Management | Communications |
|---|---|---|
| <ul style="list-style-type: none">▣ Title, description and navigation▣ Versioning settings▣ Advanced settings | <ul style="list-style-type: none">▣ Delete this document library▣ Save document library as template▣ Permissions for this document library▣ Manage checked out files▣ Workflow settings | <ul style="list-style-type: none">▣ Incoming e-mail settings▣ RSS settings |

Step 4 - Configure *DocumentBurster*

Configuring *DocumentBurster* email for distributing reports to SharePoint isn't different than the usual *DocumentBurster* email configuration

The screenshot shows the 'Configuration' page for 'My Reports -> Email'. The left sidebar contains a tree view with 'My Reports' expanded, showing sub-items: 'General', 'Email' (selected), 'Attachments', 'Upload', 'Quality Assurance', and 'Advanced'. The main content area has two tabs: 'Connection Settings' and 'Email Message'. The 'Email Server' section under 'Connection Settings' contains the following fields:

| Field | Value |
|--------------------|-------------------|
| From Name | Sender |
| From Email Address | sender@sharepoint |
| Host | sharepoint |
| User Name | sender |
| Password | •••••••• |
| Port | 25 |

Below the fields are two checkboxes: 'SSL Enabled' and 'TLS Enabled', both of which are unchecked.




documentburster@sharepoint email address was previously defined in SharePoint as being the email address of *Shared Folders* document library

The screenshot shows the 'Configuration' page for 'My Reports -> Email' with the 'Email Message' tab selected. The left sidebar is the same as in the previous screenshot. The main content area shows the 'Email Message' section with the following field:

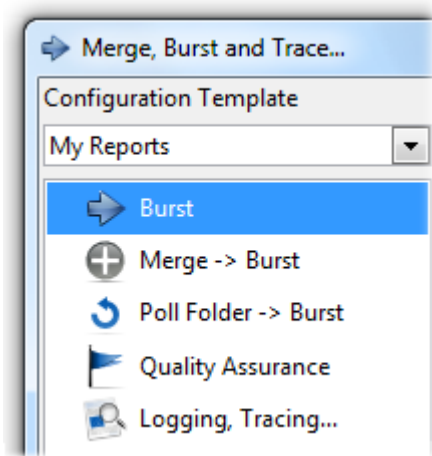
| Field | Value |
|-------|----------------------------|
| To | documentburster@sharepoint |

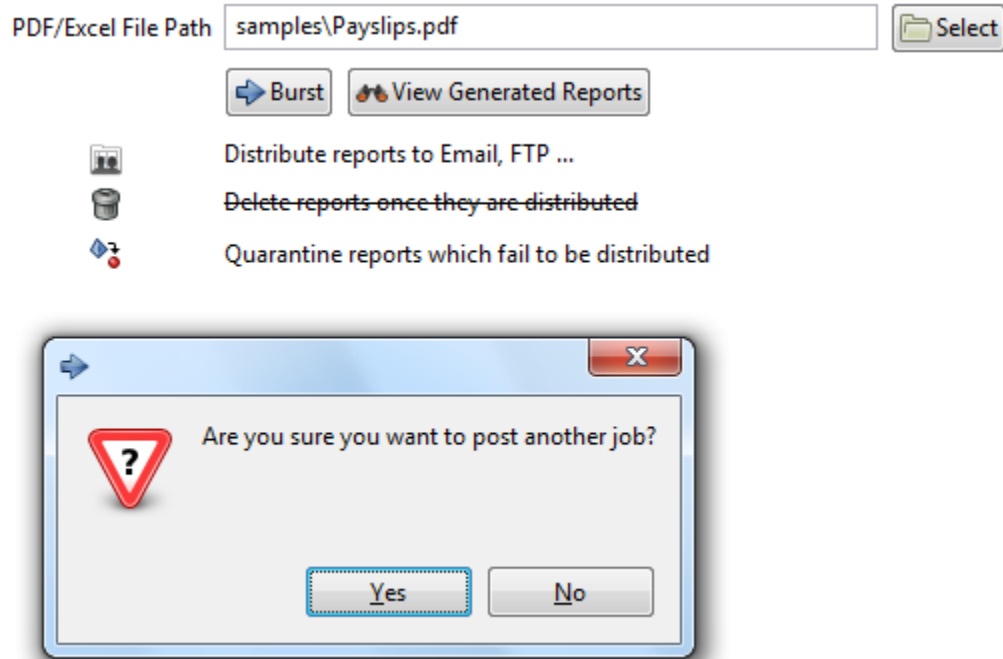
Step 5 - Burst the sample report

Double check that the email distribution is enabled

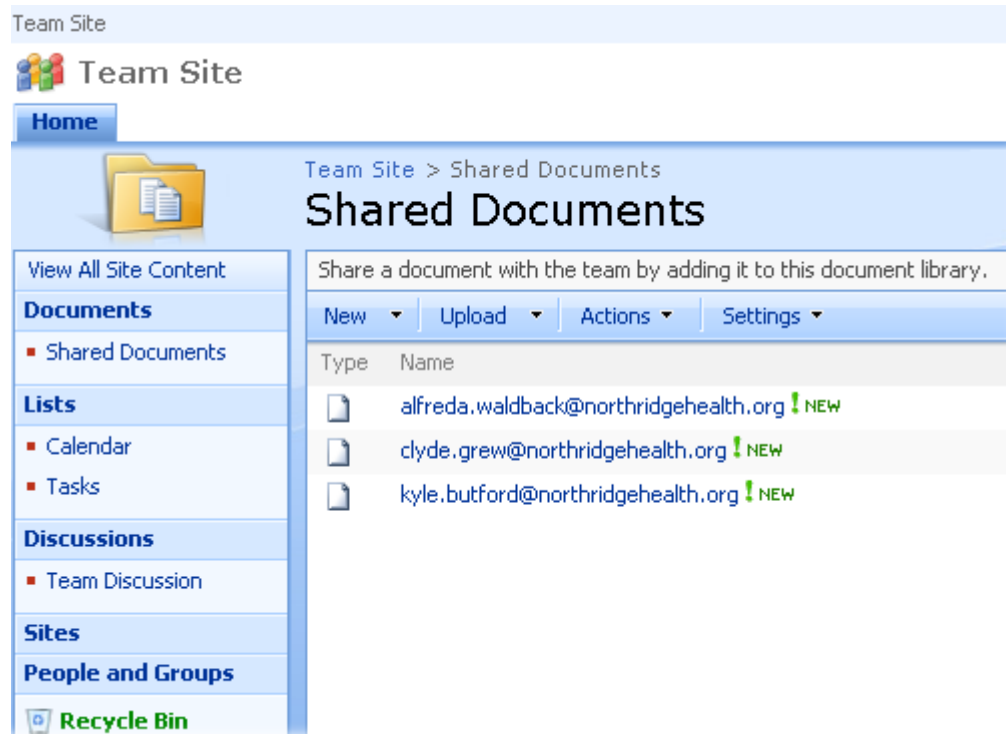
| | |
|---|---|
| Burst File Name | <input type="text" value="\$burst_token\$.input_document_extension\$"/> |
| Default Merge File Name | <input type="text" value="merged.pdf"/> |
| Output Folder | <input type="text" value="output/.input_document_name\$/.now;format=" yyyy.mm"=""/> |
| Backup Folder | <input type="text" value="backup/.input_document_name\$/.now;format=" yyyy.mm"=""/> |
| Quarantine Folder | <input type="text" value="quarantine/.input_document_name\$/.now;format=" yyyy."=""/> |
| <i>(should be a local folder)</i> | |
| Poll Folder | <input type="text" value="poll"/> |
| <i>(should be an existing local folder)</i> | |
|  | <input checked="" type="checkbox"/> Distribute reports to Email, FTP ... |
|  | <input type="checkbox"/> Delete reports once they are distributed |
|  | <input checked="" type="checkbox"/> Quarantine reports which fail to be distributed |

Select and burst samples/Payslips.pdf report





The output burst reports are uploaded to SharePoint



Email address validation

The last thing to be mentioned is about how *DocumentBuster* is validating the email addresses. This is particularly important to be noted here because, if the above steps are followed exactly, then *Document-*

Burster will not distribute any document to SharePoint. *DocumentBurster* will display, in the log file, a message similar with

Invalid TO email address 'documentburster@sharepoint'. Defining 'validatemailaddresses' configuration with the value 'false' might result in a less strict email address validation!

In the default configuration *DocumentBurster* is validating all the email addresses to be in the format *string@otherstring.ext* and will throw the above exception for any address which is not in this format (i.e. *documentburster@sharepoint*).

Strict email address validation is required for production deployments, where all email addresses should be defined as per the standard format and this is the default behavior of *DocumentBurster*. However, in the pre-production phases, when *DocumentBurster* is tested locally there might be cases when email validation should be less strict (i.e. *documentburster@sharepoint*).

Please read the section called “Advanced Settings” to see how to configure *DocumentBurster* to do less *Strict Email Address Validation*. By unchecking the *Strict Email Address Validation* configuration will make the test to pass for *documentburster@sharepoint* email address and thus the output reports will be delivered to this SharePoint address.

In *DocumentBurster* main window select Actions , Configure , Advanced



☐ **Strict Email Address Validation**

Note 1

Check back the *Strict Email Address Validation* configuration when deploying *DocumentBurster* in production.

Note 2

The scope of the above paragraphs is to give a good picture about how SharePoint can be configured to accept incoming emails from *DocumentBurster*. For complete details on how to configure SharePoint inbound email feature, please check the official Microsoft SharePoint documentation.

Note 3

If required, your organization's SharePoint administrator should be able to give further help in configuring the system.

Appendix C. *DocumentBurst* Performance

The first part of this chapter presents a general picture about report distribution performance and some high level performance guidelines for getting an increased throughput from the *DocumentBurst* software.

The last section shows a simple, yet powerful, *DocumentBurst* tool for monitoring various report bursting and report delivery performance metrics.

Performance Guidelines - Getting the Most From Your *DocumentBurst* System

DocumentBurst hardware requirements will depend on the volume and the size of the reports which the system is expected to process and distribute. The system can scale in order to support the most demanding report distribution performance requirements.

In most of the situations, *DocumentBurst* should perform well enough on any low-end machine which can decently run Windows XP. However, if your enterprise organization is expected to process really huge reports, it is advisable to allocate a dedicated and powerful machine.

Following are the hardware devices which have an influence on the overall system throughput

- **Network performance** - The quality of network and the size of the files transferred will have a considerable impact on the report distribution throughput. A good network bandwidth together with low network latency will have a positive impact when distributing a lot of reports by either email or other network based protocols.
- **CPU, Disk I/O performance** - More CPU power together with a good Disk I/O performance will improve the report bursting/separation throughput. *DocumentBurst* will benefit when running on a modern multi-core server machine.
- **Memory** - By default, *DocumentBurst* starts with 128MB and *DocumentBurst Server* starts with 256MB. The default memory configuration should be enough when processing really big reports (thousands of pages) and, if required, the memory can be further increased in order to process even bigger reports.

Minimize the network overhead

The most common scenario will be for *DocumentBurst* to distribute the reports by email. When a big number of reports is required to be distributed in a short period of time, deploying *DocumentBurst* software on the same physical machine with the email server (e.g. Microsoft Exchange) will mostly save the network overhead and will be like a boost for the overall report distribution throughput.

If possible, for email distribution performance reasons, install *DocumentBurst* on the same computer with the email server software (e.g. Microsoft Exchange).

Minimize intermediate layers

Any intermediate layer which sits in-between *DocumentBurst* and the email server software will pose a performance penalty when the reports are being distributed. For example, a common situation will be with various antivirus/firewall (e.g. Symantec, McAfee) kind of software which might monitor the network

transfer from *DocumentBuster* to the email server (e.g. Microsoft Exchange). Such intermediate layers will slow down the network performance and, if email sending performance is a critical requirement, the IT administrator of your organization should check for various approaches to minimize the performance cost of these intermediate layers. The simplest approach would be to decide that all the emails which are sent by *DocumentBuster* are legitimate and do not require additional validation. Another approach might be to configure the antivirus software to do the minimum amount of scanning necessary to meet the security requirements from your organization.

Running *DocumentBuster* in a virtual machine

DocumentBuster can be deployed and used in a virtual machine. The software is confirmed to run in Oracle VM VirtualBox and Microsoft Hyper-V and should work properly on other virtualization software too. Like any other software, there is a performance penalty when running *DocumentBuster* in a virtual machine as compared with running the software in a native operating system. The network quality, the CPU, Disk I/O and memory performance are the factors which influence the total throughput of the *DocumentBuster* software when running inside a virtual machine.

Performance Monitoring

DocumentBuster has a simple, yet powerful, mechanism to monitor various report bursting and report distribution performance metrics. This mechanism can be enabled for tracking, on an ongoing basis, the report bursting and report distribution performance during production.

DocumentBuster monitoring system displays the most relevant report bursting and report distribution performance indicators with the help of various graphs which are easy to understand and visualize.

Further more, *DocumentBuster* is publishing the performance metrics in two additional ways which can be used when integrating with professional monitoring tools

- *Log file* - *DocumentBuster* generates a relatively simple to read (and simple to parse) log file which contains aggregated performance statistics. When compared with the graphs, the log file contains a bigger set of performance metrics (at the cost of being less easy to read) and, if required, can be automatically polled and parsed for raw performance data by other third party monitoring tools.
- *JMX* - *DocumentBuster* exposes performance metrics through JMX so that monitors can be set up to alert when performance degrades below acceptable thresholds. Since JMX has become the standard interface for managing and monitoring Java applications, exposing *DocumentBuster* performance data through JMX opens up a wide range of functionality provided by third party monitoring applications.

Enable Performance Monitoring

In the default installation, *DocumentBuster* does not gather any performance related data. If performance monitoring is required, then *DocumentBuster* should be configured to gather, calculate and publish the performance statistics.

How to enable performance monitoring for *DocumentBuster*

DocumentBuster can be enabled to publish performance metrics by doing the modification which is marked as bold to the first line of `config/burst/internal/documentbuster.xml` configuration file.

- existing value - `<project name="documentbuster" default="documentbuster">`
- new value - see the bold change `<project name="documentbuster" default="performance-statistics">`

How to enable performance monitoring for *DocumentBurster Server*

DocumentBurster Server can be enabled to publish performance metrics by doing the modification which is marked as bold to the first line of `server/config/burst/internal/startServer.xml` configuration file.

- existing value - `<project name="server" default="start-server">`
- new value - see the bold change `<project name="server" default="performance-statistics">`

Save the configuration file(s) and run *DocumentBurster* again. Now, everytime a report is being processed, performance metrics will be published in the following two files

- `logs/performance/statistics.log` - contains the aggregated performance statistics calculated at regular time intervals (10 seconds by default)
- `logs/performance/graphs.log` - contains *Google Chart Tools* URLs with *DocumentBurster* performance charts

<http://code.google.com/apis/chart/>

Built-in *DocumentBurster* Performance Monitoring

Performance Statistics Log File

If configured as such, *DocumentBurster* will gather run-time performance statistics. The data is collected throughout the program execution and regularly, at a configurable time interval (10 seconds by default) the raw data is processed to produce and log aggregated statistics such as mean, minimum, maximum, standard deviation and transactions per second over a set time span. If monitoring is enabled, the aggregated statistics will be available in `logs/performance/statistics.log` log file.

Example of performance statistics log file

```
Performance Statistics    2011-10-27 17:40:20 - 2011-10-27 17:40:30
Tag                      Avg(ms)  Min    Max    Std Dev  Count
executescript_endParsePage.groovy    0.4      0      1      0.5      643
executescript_startParsePage.groovy  0.3      0      1      0.4      643
processPage                      14.8     4      27     7.5      643
```

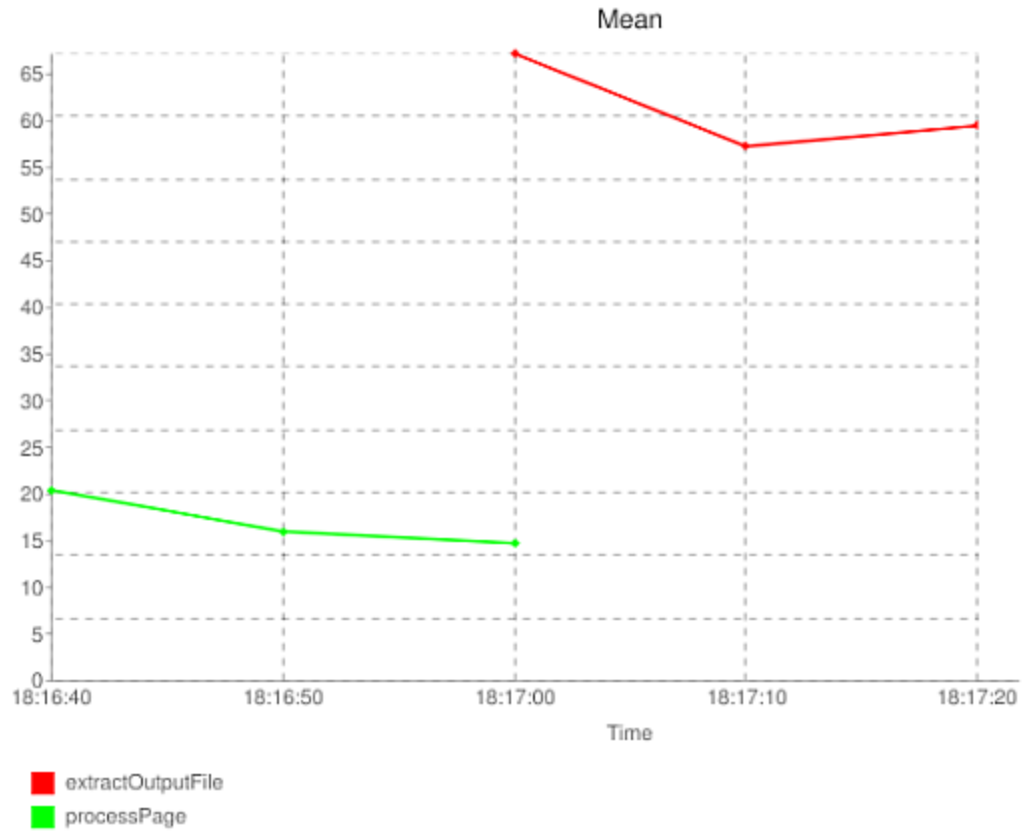
Performance Graphs

If monitoring is enabled, *DocumentBurster* will generate graphs for the average execution time and transactions per second for the following transaction types

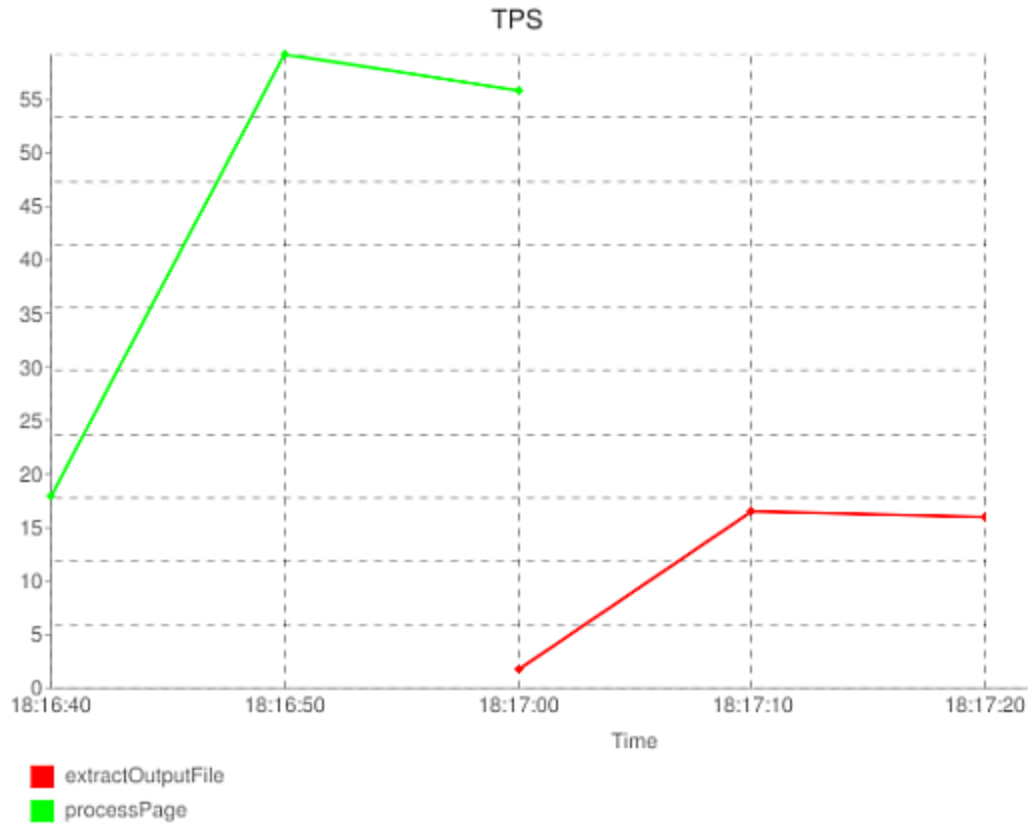
- *processPage* - transaction which is parsing the text of one page of the report for burst tokens and variable values.
- *extractOutputFile* - transaction which is extracting a separate burst report.
- *send* - transaction which is distributing the previously extracted report.

The graphs are generated as URLs to Google Chart Server in `logs/performance/graphs.log` log file.

Mean Execution Times (ms)



Transactions Per Second

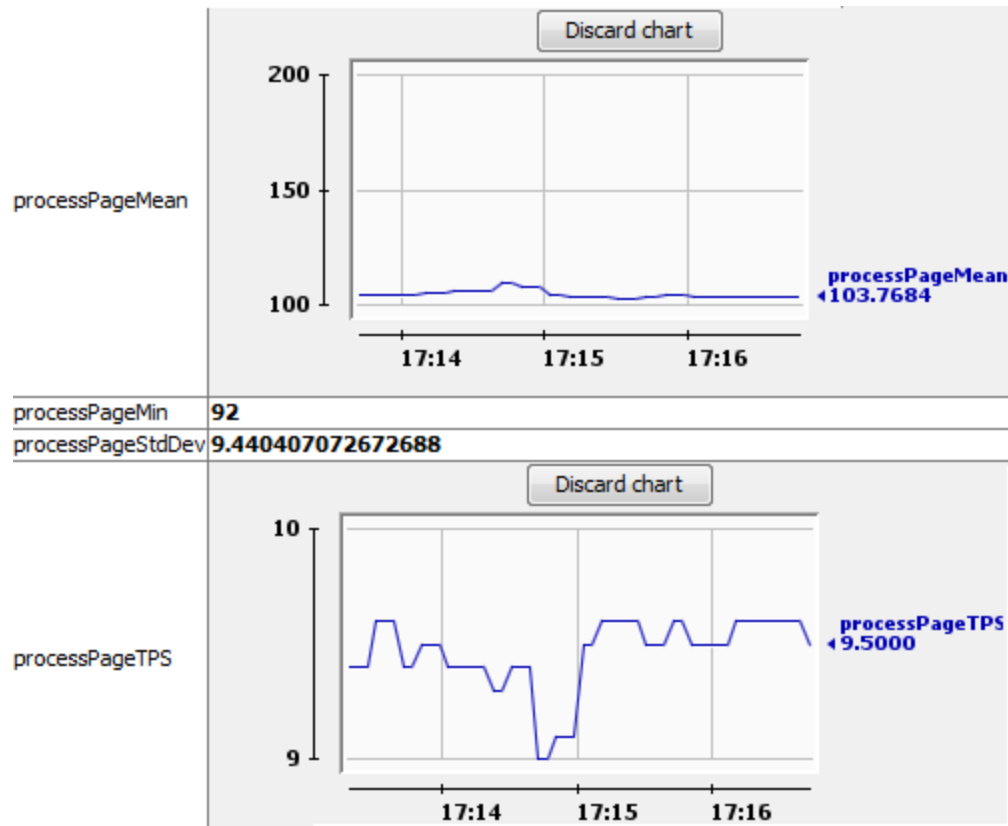


Exposing Performance Statistics as JMX Attributes

If performance monitoring is enabled, *DocumentBurster* does expose the performance data using the JMX standard. Exposing *DocumentBurster* performance data using JMX opens up a wide range of functionality, such as

- *DocumentBurster* performance metrics can be visualized using existing Java JMX consoles like JConsole or MC4J Management Console.
- Through JMX, *DocumentBurster* has the premises to be monitored using professional network monitoring software like Nagios or OpenNMS.

Visualize report bursting and report distribution performance metrics using JConsole



Integration with Professional Third Party Monitoring Tools

If required, for more advanced monitoring scenarios, *DocumentBuster* has the technical premises to be integrated with other professional monitoring tools. Such an advanced monitoring tool will allow, for example, to define performance thresholds on *DocumentBuster* metrics and to raise an alert whenever the minimum performance thresholds are not met. Once raised, the alert can be either displayed in a monitoring dashboard using red color or can be sent by email to one or more configured email addresses.

While *DocumentBuster's* monitoring system is not directly dependent on any, it can be integrated with most of the existing network monitoring platforms such as Nagios, Cacti, Hyperic, Zenoss, SpiceWorks, GroundWork, Zabbix or OpenNMS. If the IT department of your organization has already standardized on using an existing platform for doing system monitoring then it should be possible to add *DocumentBuster* to the list of applications which are closely monitored.

The documentation of the selected monitoring vendor should be consulted in order to find details on how to define and monitor a new application. In general, such tools will allow to fetch the required *DocumentBuster* performance data by either

- Constantly polling and parsing of the raw *DocumentBuster* performance log file - `logs/performance/statistics.log` for the required data.
- By using a JMX adaptor (provided by the third party tool), remotely connect and read the performance data which is exposed by *DocumentBuster* using JMX.

Further Reading

- *Google Chart Tools* - *DocumentBuster* is displaying the performance charts using Google Chart
<http://code.google.com/apis/chart/>
- *JMX* - *DocumentBuster* exposes performance metrics through JMX
http://en.wikipedia.org/wiki/Java_Management_Extensions
- *JConsole* - *DocumentBuster* can be monitored using JConsole
<http://java.sun.com/developer/technicalArticles/J2SE/jconsole.html>
- *Nagios documentation* - general documentation useful when monitoring *DocumentBuster* through Nagios
<http://www.nagios.org/documentation>
- *Cacti documentation* - general docs useful when using Cacti
<http://www.cacti.net/documentation.php>
- Hyperic HQ documentation
<http://www.hyperic.com/support/systems-monitoring-doc>
- *Zenoss documentation*
<http://www.zenoss.com/resources/docs>
- *Spiceworks - Free network monitoring software* - use Spiceworks for system monitoring
<http://www.spiceworks.com/free-network-monitoring-management-software/>
- *GroundWork - network and application monitoring software* - use GroundWork for system monitoring
<http://www.gwos.com/>
- *Zabbix - Monitoring software* - use Zabbix for system monitoring
<http://www.zabbix.com/>
- *OpenNMS - Monitoring software* - use OpenNMS for system monitoring
<http://www.opennms.org/>

Appendix D. Troubleshooting

Overview

This section provides troubleshooting information for common *DocumentBurster* difficulties.

If you are experiencing problems using *DocumentBurster* make sure that you have read and properly installed the software prerequisites, as described in *DocumentBurster in 5 Minutes* tutorial guide.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

Quick & Professional Support

If you need to contact *DocumentBurster* support, you can get help by sending an email to support@pdfburst.com

Common Problems

I Get Only 25 Reports

The free version of *DocumentBurster* has a *restriction of maximum 25 burst output reports* being generated and/or being distributed.

For bursting and generating an unlimited number of reports please purchase *DocumentBurster*.

<http://www.pdfburst.com/purchase.html>

Issues Running Basic Features?

If you don't know from where to start or you have problems running the basic features of the program, read *DocumentBurster in 5 Minutes* tutorial guide.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

UnsupportedClassVersionError Exception?

If on the console or log file there is an exception similar with

```
exception in thread "main" java
.lang.UnsupportedClassVersionError:test (unsupported major.minor
version 49.0)
at java.lang.ClassLoader.define(Class0(Native
Method)
at java.lang.Clas
```

This is happening when the program runs with an ancient java version (<Java v1.5). Read *DocumentBurster in 5 Minutes* document and double check the version of java which is installed on your computer.

DocumentBurster requires Java v1.6 (or greater) version in order to run.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

Sometimes the exception is still coming even after the latest Java is installed. This is happening because the old java is still installed and active on your computer.

The solution is to edit `documentburster.bat` and do the following change

```
java          -Djava.endorsed.dirs=./lib/endorsed          -cp          ./lib/burst/ant-launcher.jar
org.apache.tools.ant.launch.Launcher -buildfile ./config/burst/documentburster.xml -Darg1=%1 -Darg2=
%2 -Darg3=%3 -Darg4=%4 -Darg5=%5 -Darg6=%6
```

Do the **bold** change

```
"C:/Program      Files/Java/jre6/bin/java.exe"          -Djava.endorsed.dirs=./lib/en-
dorsed -cp ./lib/burst/ant-launcher.jar org.apache.tools.ant.launch.Launcher -buildfile ./config/burst/
documentburster.xml -Darg1=%1 -Darg2=%2 -Darg3=%3 -Darg4=%4 -Darg5=%5 -Darg6=%6
```

The proper path to the location where the latest java is installed should be provided. This change will force *DocumentBurst* to run with the latest java.

Mess With Multiple Java Installations

As a general rule, it is advisable to have a single Java installation on the machine where *DocumentBurst* is executed. Even if (theoretically) it is possible to have multiple java versions installed on the same computer, in practice this will only leave room for strange issues and will complicate the installation.

Windows - DocumentBurst.exe GUI is Failing to Start?

Was *GtkSharp* prerequisite installed before running *DocumentBurst.exe*? Read *DocumentBurst in 5 Minutes* guide to install the prerequisites required to run *DocumentBurst*.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

Windows - DocumentBurst.exe GUI Still Fails?

Sometimes a system reboot is required after *GtkSharp* is installed (mostly on Windows 7).

If required, restart your Windows machine after *GtkSharp* is installed.

Windows - DocumentBurst.exe GUI Still Fails?

Was *GtkSharp* prerequisite installed using the default values presented by the wizard, as per *DocumentBurst in 5 Minutes* guidelines?

GTK runtime should be properly exported and visible through the Windows %PATH% environment variable. If required, update %PATH% environment variable and change the position of GTK runtime location to be near the front.

Windows - DocumentBurst.exe GUI Still Fails?

Is the software executed from a shared drive? It should not.

DocumentBurst.exe GUI can only run from the local machine.

Windows - DocumentBurst.exe GUI Comes Up For a Second and Then Disappears?

This is how people will usually describe this issue

When I click on the exe file the program comes up for a second and then disappears. I tried using the software on my home PC and it's working fine.

Solution

This is happening when the *DocumentBurst* installation is missing few folders. In most of the situations only downloading and extracting again the zip file, in a different folder, will get the situation solved.

Sometimes, on rare occasions, some folders are not properly extracted from the zip file.

Burst Issue 1

If you have bursting problems, check that you have configured the burst tokens in-between brackets. For example {alfreda.waldback@northridgehealth.org} is a valid token.

Burst Issue 2

If you configured all the tokens properly and still the program is not working as expected, you can enable detailed logging. By checking the detailed log file you can understand what is going wrong.

In order to enable detailed logging, read the previous Chapter 7, *Quality Assurance* .

Variable Values Are Not Parsed Correctly?

Sometimes variables defined like <0>some value</0> and up to <9>some other value</9> might fail in getting parsed the proper values. Following is an example of the issue coming with Microsoft Access, while similar behavior might be observed with other report writers too.

The Problem - Example of the problem coming with Microsoft Access

I am using various MS Access reports to grab variable data using <0> text </0> . If I use a label for the text and key it into the text box as <0> report id 100 </0> it works fine but if I drop a field onto the report and then put the <0> and </0> in front and back of the field, it does not work.

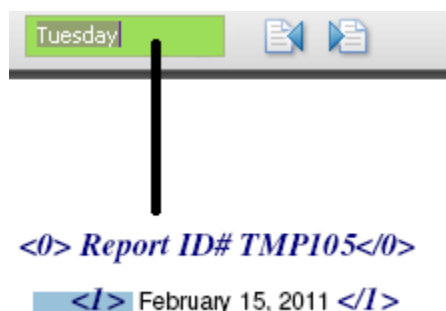
The solution - And here is the solution for the previous MS Access behavior

When you drop the fields into an MS Access report you need to define any field you use as a variable as a single field by concatenation. For example, let's say I have a field named "date" and place it on the report with a text box of <0> in front and then place a text box of </0> at the end. This will not work. You need to create one field (object) as follows: =" <0> "text" </0> ". Now it will work.

More details

If the start and end tags (e.g. <0> and </0>) are statically defined, while the content inside is a dynamic field /report formula (string value which can grow in length), the dynamic content will grow and will start to overlap with the static tags (e.g. <0> and </0>). This might cause problems when *DocumentBurst* is

parsing the variable values. See the following screenshot in which "Tuesday" hidden text was generated by a date field/formula which expanded its length and started to overlap the start `<I>` tag. In this case the text which is extracted by *DocumentBurst* is a messy `Tues<dal>y` and as a result the variable value is not properly parsed. The solution to this problem was described in the previous paragraph.



Windows - *DocumentBurst* Server Is Failing to Start?

When `server/startServer.bat` script is executed it is flashing up the cmd box and then it disappears.

Solution

- Are all the prerequisites in place? Read *DocumentBurst in 5 Minutes* tutorial guide and check all the prerequisites required for running *DocumentBurst*.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

- *DocumentBurst* should run on Java version v1.6 or greater (`java -version` MS-DOS command should return v1.6 or greater)
- If required, start the server again. Did you shut the server properly from previous runs by using `server/shutServer.bat` script?

Windows - *DocumentBurst* Server Is Not Processing the Web Console Jobs?

Solution

- Are all the prerequisites in place? Read the *DocumentBurst in 5 Minutes* tutorial guide and check all the required prerequisites.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

- *DocumentBurst* should run on Java version v1.6 or greater (`java -version` MS-DOS command should return v1.6 or greater)

Windows - *DocumentBurst* Web Console Is Failing to Start?

The link `http://localhost:8080/burst` is not working on the local machine. When `web-console/startConsole.bat` script is executed it is flashing up the cmd box then it disappears.

Solution

- Are all the prerequisites in place? Read *DocumentBuster in 5 Minutes* tutorial guide and check all the required prerequisites.

<http://www.pdfburst.com/report-bursting-quickstart.pdf>

- *DocumentBuster* should run on Java version v1.6 or greater (*java -version* MS-DOS command should return v1.6 or greater)
- Was *DocumentBuster Server* console started before, using *server/startServer.bat* script? *DocumentBuster Server* console should be started before the web console is started.
- Before starting the web console, at least one of *%JAVA_HOME%* or *%JRE_HOME%* environment variables should be properly defined on the system (either *echo %JAVA_HOME%* MS-DOS command should return a proper JDK 1.6 installation path or *echo %JRE_HOME%* MS-DOS command should return a proper JRE 1.6 installation path).

If required, on Windows, *JRE_HOME* environment variable can be manually defined in *Control Panel* -> *System properties* (WinKey + Pause) -> *Advanced* -> *Environment Variables*, e.g. C:/Program Files/Java/jre6

Windows Service Stops on Windows Server 2003?

The following problem is happening only on Windows Server 2003.

The system was tested and everything works fine on Windows Server 2008 or Windows 7.

The Problem - *DocumentBuster Server* service always stops when the system is logged off.

Solution

This is required only for Windows Server 2003.

Before doing any change make sure that

- Both *DocumentBuster Web Console* and *DocumentBuster Server* Windows services are stopped
- Both *DocumentBuster Web Console* and *DocumentBuster Server* are properly shut down using the corresponding *shutConsole.bat* and *shutServer.bat* scripts

-Xrs switch should be added in two (2) places

- *server/startServer.bat*
- *server/config/burst/internal/documentbuster.properties*

Change 1. *server/startServer.bat*

```
java -DDOCUMENTBURSTER_HOME=%CD% -Djava.endorsed.dirs=lib/endorsed -cp lib/batch/ant-launcher.jar org.apache.tools.ant.launch.Launcher -buildfile config/burst/internal/startServer.xml -emacs
```

Do the **bold** change

```
java -Xrs -DDOCUMENTBURSTER_HOME=%CD% -Djava.endorsed.dirs=lib/endorsed -cp lib/batch/ant-launcher.jar org.apache.tools.ant.launch.Launcher -buildfile config/burst/internal/startServer.xml -emacs
```

Save the file.

Change 2. `server/config/burst/internal/documentburster.properties`

`SERVER_JVM_OPTS=-XX:MaxPermSize=256m -Xms512m -Xmx512m`

Do the **bold** change

`SERVER_JVM_OPTS= -Xrs -XX:MaxPermSize=256m -Xms512m -Xmx512m`

Save the file.

Restart both *DocumentBuster* Windows services and check that *DocumentBuster* works properly when the machine is logged off.

Email Is Failing?

If you have problems in getting the email working

- Double check the email server connection details
- If your organization is using Microsoft Exchange as email server and Microsoft Outlook as email client, in this situation, you will need to give the same email server connection details which are already configured in your Microsoft Outlook email client software.

Are you running Anti-Virus software?

If your organization has any firewall and/or antivirus software (e.g. Symantec, McAfee, etc.) which is configured in between *DocumentBuster* software and the email server software (e.g. Microsoft Exchange), in this case, the antivirus and/or firewall might need to be properly configured to allow *DocumentBuster* sending emails as a good and trustable citizen.

Ask for Help

A network or IT administrator from your organization should be able to assist if you find further difficulties in configuring the email settings.

Email Still Fails?

Check the logs on the email server side. If your organization is using Microsoft Exchange as an email server, check the Microsoft Exchange log entries to see if the email server was reached and what problem was encountered.

Email SMTP Error 421

If on the console or log file there is an exception similar with

Caused by: javax.mail.MessagingException: Could not connect to SMTP host: host-here, port: port-here, response: 421

This represents *Email SMTP Error 421* (see the SMTP error code 421 in the exception) and you will need to work together with your IT Network or Microsoft Exchange administrator which should further read and interpret the email server log entries (e.g. Microsoft Exchange logs).

Possible ISP limitation

In addition, you might need to let your ISP know that you have a legitimate reason for sending many emails.

More details about *Email SMTP Error 421*

SMTP Error 421: The Mail transfer service is unavailable because of a transient event. SMTP reply 421 can be caused by many things but generally indicates that the mail server which returns this status code is currently unavailable but may be available later. For example, the server administrator may have stopped the mail service to troubleshoot a problem, or the mail server is right in the middle of rebooting, or the mail server is currently processing too many incoming messages or incoming requests, etc.... Note : “Mail Server” in this case can be any of the mail servers on the message’s route – the sending server (your server), the ISP SMTP server, or the recipient’s mail server. Clearly, if you repeatedly receive an SMTP status 421 then the problem is no longer of a transient nature and you need to investigate or inform the relevant network administrator, ISP tech support, or the recipient.

SMTP Response 421 can also be received as a result of your message server sending an email where the total number of TO, CC, and BCC users results in a number of simultaneous SMTP connections that is in excess of the number of connections your ISP or SMTP service allows. A typical error message for this situation would be : “421 Too many concurrent SMTP connections from this IP address; please try again later”. Typically, when this happens your server will have sent some of the messages (note that for all servers, each email sent by a user always gets broken down into individual separate emails to each of the recipients in the TO, CC, and BCC fields), and will automatically retry a little later to send the remaining messages.

Email SMTP Error 550 5.7.1 Unable to relay

This is how people will usually describe this issue

We are able to burst reports to colleagues with internal email addresses successfully. Our problem is that when we try to send TO or CC to external clients / emails the burst email always fails.

If on the console or log file there is an exception similar with

*Caused by: javax.mail.SendFailedException: Invalid Addresses;
nested exception is:
com.sun.mail.smtp.SMTPAddressFailedException: 550 5.7.1 Unable to relay*

In *DocumentBurster* main window select Actions , Configure , Email , Connection Settings

| | | |
|--------------------------------------|---------------------------------|---|
| User Name | <input type="text"/> |  Variables |
| Password | <input type="password"/> |  Variables |
| Port | <input type="text" value="25"/> |  Variables |
| <input type="checkbox"/> SSL Enabled | | |
| <input type="checkbox"/> TLS Enabled | | |

User name field should not be empty. It should be a valid Exchange/NT user name and the user name should be complete, including the network prefix (e.g. EHNET\cjohn).

When sending emails to external addresses, for security/spam reasons, Exchange is more strict and requires people to authenticate. This is not the case when sending emails to internal email addresses. This is the meaning of **550 5.7.1 Unable to relay** Exchange exception.

Few Emails Are Going and Then Email Stops Working

This is how people will usually describe this issue

The first pass ran through fine and files burst into folder – all 187. I then chose to email and burst - the file got through to number 45 or so and then started to fail.

In the `logs/documentburster.log` file there is an exception similar with

```
Caused by: javax.mail.MessagingException: Exception
reading response;
nested exception is:
java.net.SocketTimeoutException: Read timed out
at
com.sun.mail.smtp.SMTPTransport.readServerResponse(SMTPTransport.java:2153)
```

This is usually happening when the email SMTP server (Exchange?) your organization is using needs some tuning/configuration in order to distribute many emails in a short period of time.

Following possible approaches can be taken in order to solve the situation

- Catch up with your organization's network/Exchange administrator and he (she) should be able to assist with this situation.
- Install *DocumentBuster* on the same machine with your SMTP email server. This should eliminate most of the existing network latency.
- Is there any antivirus/firewall kind of software which is sitting in-between *DocumentBuster* and Exchange Server? If yes, then you might consider to temporarily disable the antivirus/firewall software and see if this makes any difference. If this is the problem then your organization's network admin should find the proper firewall configuration which will allow *DocumentBuster* to send emails like a good and trust-able citizen (through the firewall).

Email Is Still Failing?

DocumentBuster can be configured to log more details about the SMTP communication.

Edit the file

`config/burst/settings.xml`

and do the **bold** change

```
<emailserver><debug>false</debug></emailserver>
```

change to

```
<emailserver><debug> true </debug></emailserver>.
```

Run the program again and check *DocumentBuster* log file to see more details about the email SMTP communication.

Upload Issue?

If you have problems in getting the upload functionality to work make sure the upload command is configured properly, as described in the section called “Upload Reports” .

In addition, you might check the `logs/cURL.log` file for further details about any possible upload error.

I messed up with the configuration and now the program is failing. Is it possible to restore the default configuration?

In *DocumentBuster* main window select Actions , Configuration Templates

Select edit for the configuration template which needs to be restored and copy the default values from the `config/burst/default/defaults.xml` existing file.

