

Dove User Guide

***Dove* User Guide**

Copyright © 2010-2011 Virgil Trasca

Table of Contents

1. Introduction	1
2. Distribute reports and documents	3
Email	3
Messages and templates	3
Which message is sent?	3
Email Destinations	4
File System Destinations	4
FTP, FTPS, SFTP and TFTP	5
Local Files	6
CIFS - Samba server/Windows share	6
WebDAV	6
3. Scheduling	7
Configuration	7
4. <i>Dove</i> web console	8
5. Advanced Settings	12
FTP Client Config	12
FTPS SSL/TLS settings	12
6. Linux/UNIX	15
7. Auditing & Tracing	16
Logging	16
8. Troubleshooting	18

List of Tables

5.1. FTP/FTPs settings 13

Chapter 1. Introduction

Dove is a free software which can be used to distribute documents and reports to a variety of destination types. A typical usage scenario is to install *Dove* as a common document distribution module for existing systems. Programs written in any language can post **send document** commands for *Dove* to execute.

Currently following destination types are supported:

- Local Files
- Samba, Windows network shared drives
- Email
- FTP, TFTP, SFTP, FTPS
- WebDAV - used to distribute documents to enterprise portals such as Microsoft SharePoint

Dove can be used:

- With a **command line** interface - useful to automate various reports distribution tasks or to integrate document distribution into an existing scheduling platform.
- **Started as a server** - For complex scenarios when a big number of tasks should be handled or when integration with external systems is required.

In both cases you will need to define **destinations** prior to sending a document. A destination is basically the address where you want the document to be distributed and it might be a local file destination, network destination, email destination, ftp destination or WebDAV destination. All the destinations types are easy to define and once you defined the destinations then you will use the same straightforward command for sending documents to any destination. You will find more details on defining specific destination types in the following chapters.

When started as a server, *Dove* is waiting for commands by polling for new xml job files coming to the **poll** folder. Once a command is received (a new xml file is found in the **poll** folder) the server picks it up and distributes the message with the details found in the xml job file to the specified destination.

When giving a **send document** command, no matter if it is through the command line or through a job file, some arguments are passed with the command. The syntax is similar in both command line and xml job files interfaces.

Command Line

Following is the syntax for executing the program in command line:

```
dove.bat -f <file to distribute> -d <destination id> -m <message id> -v <variable values> -b
```

- -d <destination id> - Mandatory argument. Destination to send the document.
- -f <file to distribute> - Mandatory argument for all the destination types but not email. The file to send.
- -m <message id> - Optional argument to identify the email message to send.
- -v <variable values> - Optional argument to identify the variable values to replace in a template message.
- -b - Optional argument to upload the file to FTP using binary mode. By default files are uploaded to FTP in ASCII mode.

-m and -v arguments are used when distributing documents to email destinations and will be detailed more in the chapter referring to email.

This is an example of uploading a file to FTP through command line:

```
dove.bat -f C:\MyFiles\myfile.pdf -d myFTPDestination
```

XML Job File

Passing the arguments through the XML job file is almost identical. The arguments should come under <args> and </args> tags, one argument per line, having the format *argument = value* .

Here is how the previous FTP example will look in XML:

```
<args>  
f=C:\MyFiles\myfile.pdf  
d=myFTPDestination  
</args>
```

Chapter 2. Distribute reports and documents

As previously discussed, *Dove* can distribute documents to various destination types. It does this with the help of **destinations** which represents various end point addresses where the documents should be distributed. All destinations are configured in the file `dove/config/dove-destinations.xml`.

Email destination are defined using the well known To, Cc, Bcc address fields, while the other file systems destination types are URI based.

In the folder `dove/samples` you can find sample XML job files which you can be used as reference.

Email

When it comes about sending mails, *Dove* gives good options like:

- **Send an email message without any file attached** - for doing this you will skip the `-f` option when giving the arguments.
- **Send email message with a file attached** - you will specify the `-f` option when giving the arguments.
- **Use a default email message** - you don't have to specify again and again the same message.
- **Use message templates** - you can customize the message using templates and variables.

Messages and templates

All email messages are configured in the file `dove/config/dove-email-messages.xml`. An email message can be defined with the exact subject and text or it can be defined as a message template.

Templates

Message templates are useful when you need few common messages to be customized based on the recipient where the message will be sent. A message template should contain the variable parts in between `$` characters. The exact values of the variables will be given when the message will be sent through the `-v <variable values>` argument. Variables values should be in the following syntax `var_name:var_value` and separated by `;`

Example

For example a template like `Hello $name$, Cheers from $sender$!` being processed with a `-v name:John;sender:Dove` argument will generate the message `Hello John, Cheers from Dove!`

Which message is sent?

Following rules are used to determine which message to send:

Procedure 2.1. Email message rules

1. `-m <message id>` - message specified explicit using `-m` argument is highest in priority.

2. **message** attribute used when the email destination is defined in `dove-destinations.xml` - second in priority.
3. **subject/text** specified inline when the destination is defined in `dove-destinations.xml` - third in priority.
4. **default message** defined in the `dove-email-messages.xml` - last in priority. There can be maximum one default message defined in `dove-email-messages.xml` file and if defined it should have `id="default"`.

Email Destinations

The simplest way to send an email is using the following command:

```
dove.bat -d john.george@mail.ru
```

This command will send an email to a single email address `john.george@mail.ru`. This may be enough for most of the situations but sometimes you might need to send an email to multiple recipients. For sending an email message to multiple recipients you should define an email destination. All the destinations should be defined in the file `dove/config/dove-destinations.xml`.

These are the elements of an email destination:

```
<destination id="destination1" message="msg1">

<to>to1@email.address;to2@email.address</to>
<cc>cc1@email.address;cc2@email.address</cc>
<bcc>bcc1@email.address;bcc2@email.address</bcc>

<subject>Sample Dove Subject</subject>
<text>Sample Dove Text</text>

</destination>
```

- `<to>`, `<cc>`, `<bcc>` should have the email recipients separated by `;` or `,`
- The **id** attribute is mandatory.
- The green elements are optional. `subject/text` are used to specify inline an email message/template while `message` attribute is used to specify an external email message/template which should be defined in the `dove/config/dove-email-messages.xml` configuration file.

File System Destinations

These are the elements of a file system based destination:

```
<destination id="destination2">
    <uri>put a valid URI here</uri>
</destination>
```


- The **id** attribute is mandatory.
- The **uri** should be valid corresponding with the type of the created destination.

FTP, FTPS, SFTP and TFTP

Four different FTP destination types are supported: classic FTP, FTPS, SFTP and TFTP.

FTP

Provides access to the files on an FTP server.

URI Format

`ftp://[username [: password]@] hostname [: port][absolute-path]`

Examples

`ftp://myusername:mypassword@somehost/pub/downloads/somefile.tgz`

FTPS

Provides access to the files on an FTP server using FTPS protocol.

URI Format

`ftps://[username [: password]@] hostname [: port][absolute-path]`

Examples

`ftps://myusername:mypassword@somehost/pub/downloads/somefile.tgz`

TFTP

Provides access to the files on an TFTP server.

URI Format - Same as ftp but without user and password

`tftp://:@hostname [: port][absolute-path]`

Examples

`tftp://:@somehost/pub/downloads/`

SFTP

Provides access to the files on an SFTP server (that is, an SSH or SCP server).

URI Format

`sftp://[username [: password]@] hostname [: port][absolute-path]`

Examples

`sftp://myusername:mypassword@somehost/pub/downloads/somefile.tgz`

Local Files

Provides access to the files on the local physical file system.

URI Format

[file://] absolute-path

Where absolute-path is a valid absolute file name for the local platform. UNC names are supported under Windows.

Examples

- file:///home/someuser/somedir
- file:///C:/Documents and Settings
- file://///somehost/someshare/afile.txt
- /home/someuser/somedir
- c:\program files\some dir
- c:/program files/some dir

CIFS - Samba server/Windows share

Provides access to the files on a CIFS server, such as a Samba server, or a Windows share.

URI Format

smb://[username [: password]@] hostname [: port][absolute-path]

Examples

smb://somehost/home

WebDAV

Provides access to files on a WebDAV server.

URI Format

webdav://[username [: password]@] hostname [: port][absolute-path]

Examples

webdav://somehost:8080/dist

Note:

Apache Commons VFS is used to abstract various destination types and URI formats. You can read more about *VFS file system types here* [<http://commons.apache.org/vfs/filesystems.html>]

Chapter 3. Scheduling

Dove can handle scheduled document and report distribution jobs. By default the software can handle distribution jobs scheduled for nightly (midnight) execution. It is really easy to customize the scheduling for yourself - if you are familiar with other *cron* like schedulers than you will feel comfortable with the scheduling mechanism which is implemented in *Dove*. Yearly, monthly, weekly, daily, hourly or your own custom reports distribution schedule are all easy to define.

While for executing ad hoc, immediate document distribution jobs, *Dove* is checking the `dove/poll` folder, scheduled XML jobs should be placed in the `dove/input-files/scheduled` directory. *Dove* will properly trigger the document distribution jobs to happen at the correct date and time, depending on how the scheduling was configured.

Configuration

Dove scheduling is configured using *cron expressions*. The default scheduling is configured to trigger daily, at midnight.

At the end of the configuration file `dove/config/internal/batch-context.xml` there is an entry similar with

```
<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. You can replace this based on your specific needs to trigger the schedule yearly, monthly, weekly, daily - at different time, hourly or at any other given time. *Cron expression* documentation is out of the scope of this user guide, you can read in detail about how to configure a cron expression 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:

In order for the new value to take effect, a server restart is required after the cron expression configuration is changed.

Chapter 4. Dove web console

Dove comes with a web based console interface which can be accessed from the browser of your choice. The console can be used for uploading of new job distribution commands 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 you have installed *Dove* execute `web-console/startConsole` script to get the console started. After few seconds you can access the application by typing following URL in the browser of your choice - `http://machine_name:8080/dove/` for example `http://localhost:8080/dove/`.

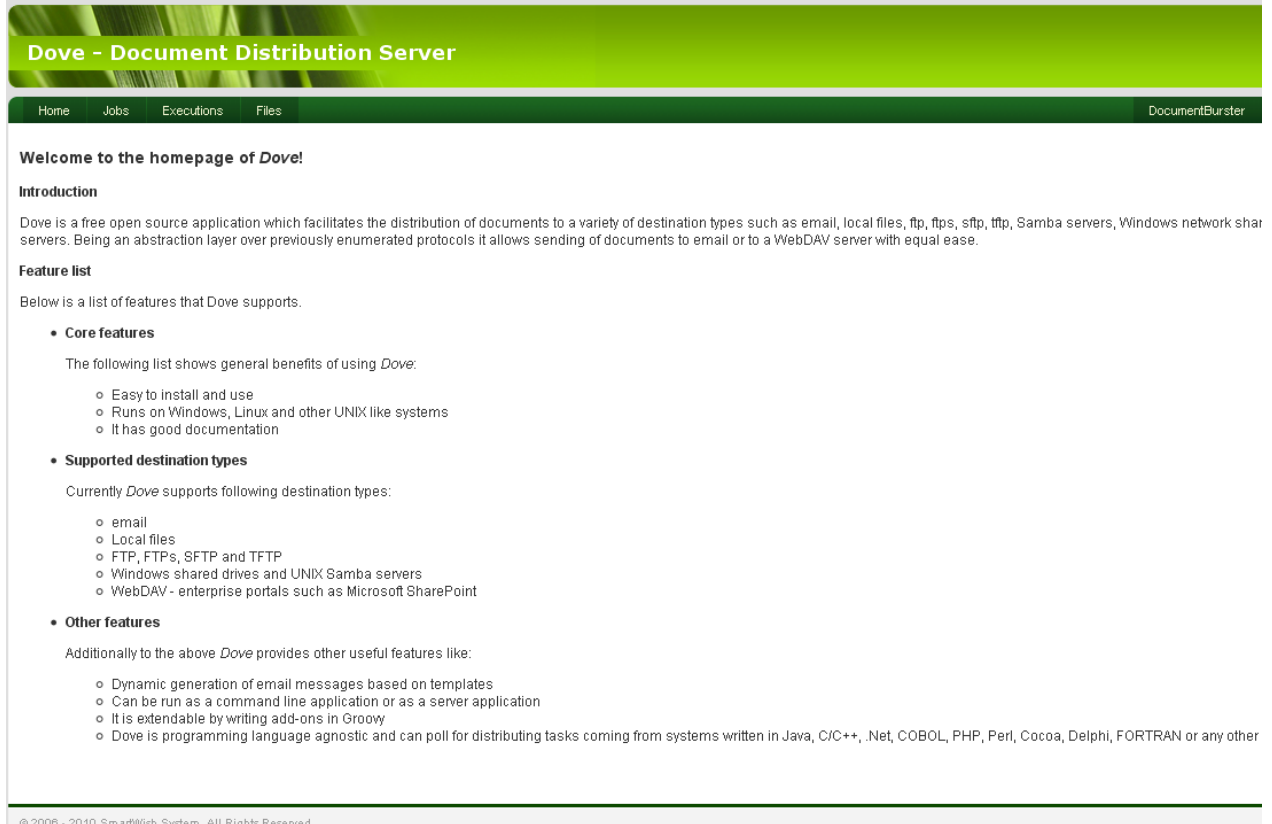
Documents and reports can be uploaded through the *Files - Upload Files* menu entry. Uploaded files will be placed in the `dove/input-files` folder with a timestamp attached to the file name. The same filename should be used as argument in the XML job file which will be uploaded.

XML job files can be uploaded through the *Files - Submit Distribution Jobs* menu entry. Uploaded job files will be placed in the `dove/poll` folder and will be automatically picked up and executed by the server. The filename argument used in the XML should be the same as the file previously uploaded to `dove/input-files` folder, including the automatically attached timestamp.

Scheduled XML job files can be uploaded through the *Files - Schedule Distribution Jobs* menu entry. Uploaded job files will be placed in the `dove/input-files/scheduled` being scheduled for execution at a later time. The filename argument used in the XML should be the same as the file previously uploaded to `dove/input-files` folder, including the automatically attached timestamp.

Dove web console can be used for submitting new jobs for immediate execution, scheduling jobs for later execution, viewing the currently running jobs and for checking the status, history and the logs of the previously submitted jobs.

Following are some screen shots from the application:



Dove - Document Distribution Server

Home Jobs Executions Files DocumentBuster

Welcome to the homepage of Dove!

Introduction

Dove is a free open source application which facilitates the distribution of documents to a variety of destination types such as email, local files, ftp, ftps, sftp, tftp, Samba servers, Windows network share servers. Being an abstraction layer over previously enumerated protocols it allows sending of documents to email or to a WebDAV server with equal ease.

Feature list

Below is a list of features that Dove supports.

- **Core features**
The following list shows general benefits of using *Dove*.
 - Easy to install and use
 - Runs on Windows, Linux and other UNIX like systems
 - It has good documentation
- **Supported destination types**
Currently *Dove* supports following destination types:
 - email
 - Local files
 - FTP, FTPs, SFTP and TFTP
 - Windows shared drives and UNIX Samba servers
 - WebDAV - enterprise portals such as Microsoft SharePoint
- **Other features**
Additionally to the above *Dove* provides other useful features like:
 - Dynamic generation of email messages based on templates
 - Can be run as a command line application or as a server application
 - It is extendable by writing add-ons in Groovy
 - Dove is programming language agnostic and can poll for distributing tasks coming from systems written in Java, C/C++, .Net, COBOL, PHP, Perl, Cocoa, Delphi, FORTRAN or any other

© 2006 - 2010 SmartWish System. All Rights Reserved.

Dove home page



Dove - Document Distribution Server

Home Jobs Executions Files

Job Names Registered

Name	Description	Execution Count	Launchable	Incrementable
distributeJob	No description	1	false	false

Rows: 1-1 of 1 Page Size: 20

© 2006 - 2010 SmartWish System. All Rights Reserved.

Jobs page



Dove - Document Distribution Server

Home Jobs Executions Files

Recent and Current Job Executions

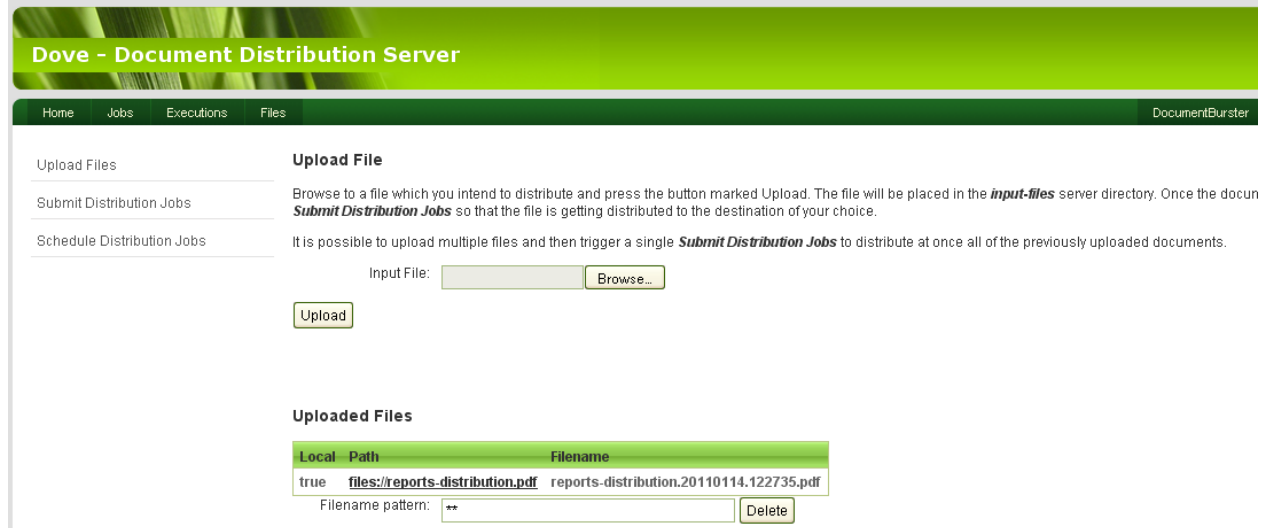
Stop All

ID	Instance	Name	Date	Start	Duration	Status	ExitCode
0	0	distributeJob	2011-01-07	13:13:18	00:00:05	COMPLETED	COMPLETED

Rows: 1-1 of 1 Page Size: 20

© 2006 - 2010 SmartWish System. All Rights Reserved.

Jobs executions page - View status and history of distribution jobs



Dove - Document Distribution Server

Home Jobs Executions Files DocumentBuster

Upload Files
Submit Distribution Jobs
Schedule Distribution Jobs

Upload File

Browse to a file which you intend to distribute and press the button marked Upload. The file will be placed in the *input-files* server directory. Once the document is uploaded, you must click the **Submit Distribution Jobs** button so that the file is getting distributed to the destination of your choice.

It is possible to upload multiple files and then trigger a single **Submit Distribution Jobs** to distribute at once all of the previously uploaded documents.

Input File: Browse...

Upload

Uploaded Files

Local Path	Filename
files:/reports-distribution.pdf	reports-distribution.201110114.122735.pdf

Filename pattern: Delete

Upload files page - Uploaded files will be placed in the **dove/input-files** folder

The screenshot shows the 'Dove - Document Distribution Server' web console. The header is green with the title 'Dove - Document Distribution Server' and navigation links for 'Home', 'Jobs', 'Executions', 'Files', and 'DocumentBuster'. The main content area is divided into a left sidebar with links for 'Upload Files', 'Submit Distribution Jobs', and 'Schedule Distribution Jobs', and a main panel titled 'Upload XML job files'. The main panel contains instructions: 'Browse to a XML job distribution file and press the button marked Upload. Once the job file is uploaded it will be picked up by the server which will execute 1 jobs.' and 'One XML file can contain multiple document distribution jobs.' Below this is a form with an 'XML job file:' label, a text input field, and a 'Browse...' button. An 'Upload' button is positioned below the form. At the bottom, it states 'There are no files to display.'

Upload jobs - Uploaded jobs will be picked up and executed by the server

The screenshot shows the 'Dove - Document Distribution Server' web console. The header is green with the title 'Dove - Document Distribution Server' and navigation links for 'Home', 'Jobs', 'Executions', 'Files', and 'DocumentBuster'. The main content area is divided into a left sidebar with links for 'Upload Files', 'Submit Distribution Jobs', and 'Schedule Distribution Jobs', and a main panel titled 'Schedule XML job files'. The main panel contains instructions: 'Browse to a XML job distribution file and press the button marked Upload. Once the job file is uploaded it will be scheduled by the server for nightly execution distribution jobs.' and 'One XML file can contain multiple document distribution jobs.' Below this is a form with an 'XML job file:' label, a text input field, and a 'Browse...' button. An 'Upload' button is positioned below the form. At the bottom, it states 'There are no files to display.'

Schedule distribution jobs - schedule documents distribution for executing at a later time

Chapter 5. Advanced Settings

Following are advanced settings and most of the times you will not need to change anything, the default values will work just fine. However there are cases in which you will need to fine tune *Dove* to be in synch with your specific requirements.

FTP Client Config

This section applies when configuring both **FTP and FTPS** destinations. The settings are configured in **dove/config/dove-config.xml** file. These settings support configuration of FTP servers which express file timestamps in formats and languages other than those for the US locale, which although it is the most common is not universal. Unfortunately, nothing in the FTP specification allows this to be determined in an automated way, so manual configuration such as this is necessary.

Following are the settings which can be configured:

```
<clientconfig id="sampleClientConfig">
  <systemkey></systemkey>
  <defaultdateformatstr></defaultdateformatstr>
  <lenientfuturedates>false</lenientfuturedates>
  <recentdateformatstr></recentdateformatstr>
  <serverlanguagecode></serverlanguagecode>
  <servertimezoneid></servertimezoneid>
  <shortmonthnames></shortmonthnames>
</clientconfig>
```

Please follow *Commons Net documentation* [<http://commons.apache.org/net/api-release/org/apache/commons/net/ftp/FTPClientConfig.html>] for details on each of the entry.

id attribute is mandatory and uniquely identifies the client config settings. This id will be used when defining the FTP/FTPS destinations to assign specific settings using the **optional clientconfig** attribute.

Example

```
<destination id="sampleFTPDestination" clientconfig="sampleClientConfig">
  <uri>ftp://john.george:john123@ftp.john.com/invoice_pdfs/</uri>
</destination>
```

FTPS SSL/TLS settings

Most of the following entries apply when configuring **FTPS** destinations. **passivemode** is the only entry which applies to both FTP and FTPS. The settings are configured in **dove/config/dove-config.xml** file.

Like previously pointed most of the times the default SSL settings will just work without you doing anything special. However in few cases you might need to apply what is described in the next paragraphs.

Following are the settings which can be configured:


```

<ssl id="sampleSSLConfig">
<passivemode>>false</passivemode>
<securityprotocol></securityprotocol>
<execpbsz>0</execpbsz>
<execprot>P</execprot>
<isimplicit>>false</isimplicit>
<truststorefile></truststorefile>
<truststoretype></truststoretype>
<truststorealgorithm></truststorealgorithm>
<truststorepassword></truststorepassword>
<keystorefile></keystorefile>
<keystoretype></keystoretype>
<keystorealgorithm></keystorealgorithm>
<keystorepassword></keystorepassword>
<keystorekeypassword></keystorekeypassword>
</ssl>

```

id attribute is mandatory and uniquely identifies the ssl settings. This id will be used when defining the FTP and FTPS destinations to assign specific settings using the **optional ssl** attribute.

Example

```

<destination id="sampleFTPSTDestination" ssl="sampleSSLConfig">
<uri>ftps://john.george:john123@ftp.john.com/invoice_pdfs/</uri>
</destination>

```

Table 5.1. FTP/FTPs settings

Name	Default Value	Description
passivemode	false	FTP and FTPS: Specifies whether to use passive mode connections. Default is active mode (false).
securityprotocol	TLS	FTPS only: Sets the underlying security protocol. The following values are defined: TLS: Transport Layer Security SSL: Secure Sockets Layer
execprot	P	FTPS only: This option specifies the protection level. Possible values are: C: Clear S: Safe (SSL protocol only) E: Confidential (SSL protocol only) P: Private
execpbsz	0	FTPS only: This option specifies the buffer size of the secure data channel.
isimplicit	false	FTPS only: Sets the security mode(implicit/explicit). Default is explicit (false).

Name	Default Value	Description
truststorefile	null	FTPS only: Sets the trust store file, so that the FTPS client can look up for trusted certificates.
truststoretype	JKS	FTPS only: Sets the trust store type.
truststorealgorithm	SunX509	FTPS only: Sets the trust store algorithm.
truststorepassword	null	FTPS only: Sets the trust store password.
keystorefile	null	FTPS only: Sets the key store file, so that the FTPS client can look up for the private certificate.
keystoretype	JKS	FTPS only: Sets the key store type.
keystorealgorithm	SunX509	FTPS only: Sets the key store algorithm.
keystorepassword	null	FTPS only: Sets the key store password.
keystorekeypassword	null	FTPS only: Sets the private key password.

Chapter 6. Linux/UNIX

Dove is coming with shell scripts for running the program under UNIX like environments. All the things which were discussed for Windows remain the same with the only difference being that the corresponding UNIX shell scripts should be executed under Unix OS. You can find more details in the **Chapter 4. Starting *Dove* as a server** of the quick start document.

Chapter 7. Auditing & Tracing

Logging

When dealing with documents it is important to have a good mechanism for auditing and tracing the possible problems.

Dove has good support for logging all of the activities and for tracing back the documents which failed to be distributed.

More details

The program log files can be found under the **dove/logs** folder. There are two log files **batch.log** and **dove.log**. The first file is created when the program is started as a server and is logging the events related with the jobs which are being posted to the server. Such events include when a new job file was picked up, what arguments were passed to the job, if it completed with success or with failure etc. The second log file is keeping more fine grained details about each separate distribution task and what happened with it.

By default the program is logging all the exceptions and the some informational events. This is for making the log files easier to read. If required, you can configure the program to generate much more detailed log files. To do this you have to go in **log4j.properties** file and do the following changes:

```
log4j.category.com.smartwish.dove=info, stdout,dove
log4j.category.com.smartwish.batch=info, stdout,batch
```

Do the bold changes

```
log4j.category.com.smartwish.dove=debug, stdout,dove
log4j.category.com.smartwish.batch=debug, stdout,batch
```

This change will generate more logs which can be used for tracing of possible problems.

Dove can automatically send you and email whenever some problem is happening.

It is possible to configure the program to send emails whenever a problem appears. For doing this please go in **log4j.properties** and give your email server details in the following places:

```
log4j.appender.EMAIL=org.apache.log4j.net.SMTPAppender
log4j.appender.EMAIL.SMTPHost= give your smtp host
log4j.appender.EMAIL.SMTPUsername= give your user name
log4j.appender.EMAIL.SMTPPassword= give your email server password
log4j.appender.EMAIL.From=
log4j.appender.EMAIL.To=
log4j.appender.EMAIL.subject=Dove failure
```

Dove can quarantine the files which failed to be delivered

It is crucial for the software to safely distribute all the documents to the proper destinations.

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. Whatever will be the failing reason, the program can be configured to quarantine the failed documents to a folder with the same name. This way at a later point in time you will see the failed documents and you will take a decision (either to distribute again or to do something else).

In the file `dove/config/dove-config.xml` you need to change the following entry:

```
<quarantinefiles>true</quarantinefiles>
```

Chapter 8. Troubleshooting

- If you don't know how to start or you have problems in running the basic features of the program please read the *Dove QuickStart* tutorial guide.
- If you have problems in getting the email working please double check your email server connect settings. If it's still not working than you can ask the help of your network administrator.
- If you have problems in getting FTP working, please double check you have the proper FTP uri details as described in the **FTP URI Format**. You might also need to read the **FTP Client Settings**.
- If you have problems in getting FTPs working, please double check you gave the proper FTPs uri details as described in the **FTPS URI Format**. You might also need to read the **SSL/TLS Settings**.
- For any issue please follow the program's log files as described in **Auditing & Tracing chapter**.
- For low level debug of SSL/TLS Connections you can change **dove.bat/dove.sh** or **startDoveServer.bat/startDoveServer.sh** and add one of the following switches to the java command found on the last line of the script.
 1. To view all debugging messages: **-Djavax.net.debug=all**
 2. To view the hexadecimal dumps of each handshake message:
-Djavax.net.debug=ssl:handshake:data
 3. To view the hexadecimal dumps of each handshake message, and to print trust manager tracing:
-Djavax.net.debug=SSL,handshake,data,trustmanager
- If you don't find an answer to your questions in the documentation, there is also **Dove Forum Help available here**. [<http://sourceforge.net/projects/doveman/forums/forum/1156096>] Please feel free to ask your questions.