

# ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-net



Peter Reutemann

December 20, 2017

©2012-2016



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*



Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-sa/4.0/>

# Contents

<b>1</b>	<b>Email</b>	<b>7</b>
1.1	Global settings . . . . .	7
1.2	Actors . . . . .	8
1.3	Addressbook . . . . .	9
1.4	Command-line . . . . .	9
1.5	Support email . . . . .	9
<b>2</b>	<b>FTP</b>	<b>11</b>
<b>3</b>	<b>SSH</b>	<b>13</b>
<b>4</b>	<b>SSL</b>	<b>15</b>
<b>5</b>	<b>SMB</b>	<b>17</b>
<b>6</b>	<b>Remote commands</b>	<b>19</b>
<b>7</b>	<b>Miscellaneous</b>	<b>21</b>
	<b>Bibliography</b>	<b>23</b>



# List of Figures

1.1	Email preferences . . . . .	8
1.2	Email address book . . . . .	9
4.1	SSL preferences . . . . .	15
5.1	SMB preferences . . . . .	17
7.1	Telnet dialog . . . . .	22



# Chapter 1

## Email

Flows are ideal for being run as background jobs (“-headless” flag). For example, importing or processing data in batches can be done at night time. Of course, you want to be notified if something went wrong or some predictions are off. Adding the *SendEmail* sink to existing flows, allows for automatic sending of emails that were generated by the *CreateEmail* transformer: if everything is OK then send an email to the customer, otherwise send an email to sysadmin. The *CreateEmail* actor adds all incoming file names, e.g., the array output of a *DirectoryLister* source, as attachments before sending the email off to the specified recipients. You can also define a custom subject and body. Variables can be placed in subject and body alike, as they get expanded when creating the email.

In order to be able to send emails, ADAMS needs to know what SMTP server to connect to. The following example configures ADAMS to send emails using a Gmail account <sup>1</sup>.

There are two ways of configuring Email:

- *globally* – using the preferences
- *per flow* – using the *SMTPConnection* standalone actor

### 1.1 Global settings

For configuring email globally, use the dialog available from the main menu (*Program* → *Preferences* → *Email*), as depicted in Figure 1.1. The placeholders *USER* and *PASSWORD* have to be replaced with the actual user credentials and *YOUR NAME* with the actual user’s name, of course.

Alternatively, you can also simply create a properties file in the `$HOME/.adams` directory, called `Email.props`. The content for the Gmail setup would look like this:

```
Enabled=true
SmtpStartTls=true
SmtpPassword=
SmtpServer=smtp.gmail.com
```

---

<sup>1</sup>For more details, see the following Gmail help page:  
<https://support.google.com/mail/bin/answer.py?hl=en&answer=13287>

```

Smtptimeout=30000
Smtpport=587
Smtprequiresauthentication=true
Smtputer=USER
Defaultaddressfrom=YOUR NAME <USER@gmail.com>
Defaultsignature=

```

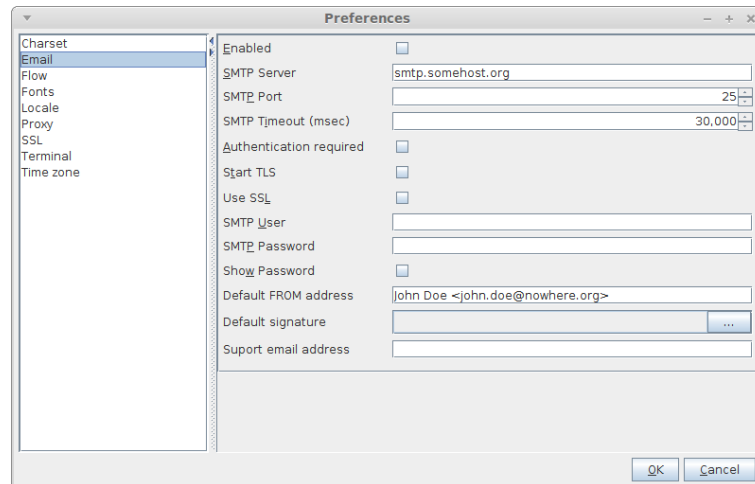


Figure 1.1: Email preferences

## 1.2 Actors

The following standalone actors are available:

- *SMTPConnection* – for configuring a SMTP server connection.

The following transformers are available:

- *CreateEmail* – creates an Email object. Interprets incoming files as attachments.
- *EmailFileReader* – reads email files with the specified email reader.

The following sinks are available:

- *EmailFileWriter* – writes Email objects to files using the specified email writer.
- *EmailViewer* – displays an Email object; can be used in conjunction with the *DisplayPanelManager* as well.
- *SendEmail* – sends Email objects to a SMTP server.

### SMTPConnection

If you do not want to store the password with the flow - after all, the password is only obscured with base64 encoding - you can enable the *promptForPassword* option. This will prompt the user with a dialog for entering a password to be used for the connection.



## 1.3 Addressbook

ADAMS also offers a very simple addressbook for emails. Figure 1.2 shows a screenshot of it.

First	Last	Email	Phone	Address	Note
John	Doe	noone@example.org	+64 1 234-5678	5 In the Wopwops Middleofnowhere	

Figure 1.2: Email address book

## 1.4 Command-line

Using the *adams.core.net.SimpleMailer* command-line tool, you can read and send previously saved emails. The following command-line loads/sends all files in the directory */some/where* that end with *.props* using the *PropertiesEmailFileReader* reader:

```
java -cp lib/* adams.core.net.SimpleMailer
  -env adams.env.Environment
  -reader adams.data.io.input.PropertiesEmailFileReader
  -watch-dir /some/where
  -regex ".*.props"
  -D 1
```

After a file got successfully loaded and sent, it gets (by default) the *.sent* extension appended. If the process failed, the *.failed* extension is used (by default). Use the *-h* or *-help* option on the command-line to list all the available options.

## 1.5 Support email

In order to make it easier for users or clients, you can set up a *support email*, which makes it easy to send through error reports. These error reports get the *system info* and current content of the *console window* automatically attached.

To enable this feature, configure the email setup through the preferences and fill in a *Support email address*. After the application has been restarted, you get a *Send error report* menu item in the *Program* menu and the *Send error report* action in the flow's notification area is enabled as well (this one also attaches the flow that generated the error for further analysis).

## Chapter 2

# FTP

In addition to Email, ADAMS also supports FTP (file transfer protocol). Since FTP does not encrypt the commands or transferred data, it is recommended to not use it outside a company's network or for non-anonymous access (e.g., download from a public FTP server).

The following actors are available:

- *FTPConnection* – standalone actor that defines the server connection.
- *FTPDisconnect* – transformer that closes an existing connection, to avoid too many open connections (simply passes through any token it receives).
- *FTPGet* – for obtaining a remote file via FTP.
- *FTPNoOp* – sends a NOOP command to the server; useful when trying to keep the connection alive.
- *FTPSend* – for sending a file via FTP to a remote host.

The *FTPConnection* standalone allows you to enter the password at runtime, if you do not want to store the password with the flow setup. At execution time, the user gets prompted with a dialog to enter a password to be used for the connection.

There are also several *searchlets* available for the *FileSystemSearch* source actor that allow the listing of remote files.



## Chapter 3

# SSH

ADAMS also comes with SSH support, thanks to the JSch library <sup>1</sup>. This allows for secure remote access. At the time of writing, RSA <sup>2</sup> and DSA <sup>3</sup> are supported for public key encryption schemes.

The following actors are available:

- *SSHConnection* – standalone actor that defines the server connection (user/password or public key authentication).
- *SSHExec* – for executing remote commands.
- *ScpFrom* – for copying a file *from* a remote host using secure copy.
- *ScpTo* – for copying a file *to* a remote host using secure copy.
- *SFTPGet* – for obtaining a remote file via secure FTP.
- *SFTPSend* – for sending a file via secure FTP to a remote host.

The *SSHConnection* standalone allows you to prompt the user as runtime to enter a password, in case you do not want to store the connection's password with the flow setup. The user has to enter the password through a dialog that pops up when the flow gets executed.

There are also several *searchlets* available for the *FileSystemSearch* source actor that allow the listing of remote files.

---

<sup>1</sup><http://www.jcraft.com/jsch/>

<sup>2</sup>[http://en.wikipedia.org/wiki/RSA\\_%28algorithm%29](http://en.wikipedia.org/wiki/RSA_%28algorithm%29)

<sup>3</sup>[http://en.wikipedia.org/wiki/Digital\\_Signature\\_Algorithm](http://en.wikipedia.org/wiki/Digital_Signature_Algorithm)



## Chapter 4

# SSL

How ADAMS handles SSL, e.g., for HTTPS connections, can be configured in the *SSL* preferences. Figure 4.1 shows a screenshot of the default settings.

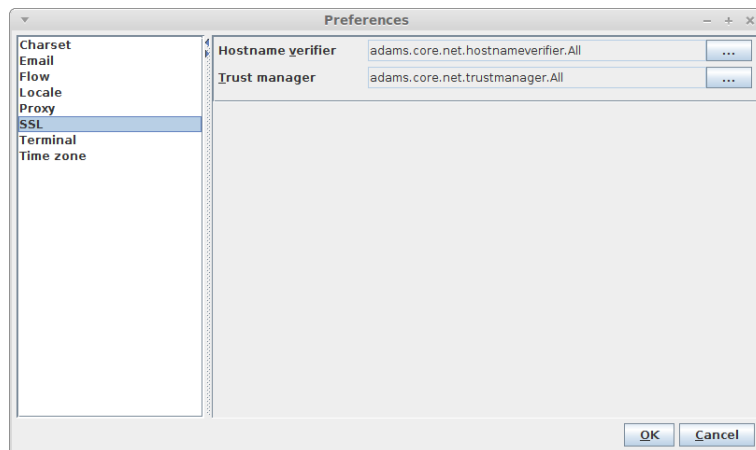


Figure 4.1: SSL preferences





## Chapter 5

# SMB

What WINS server ADAMS uses for SMB (i.e., Windows shares), can be configured in the *SMB* preferences. Figure 5.1 shows a screenshot of the default settings.

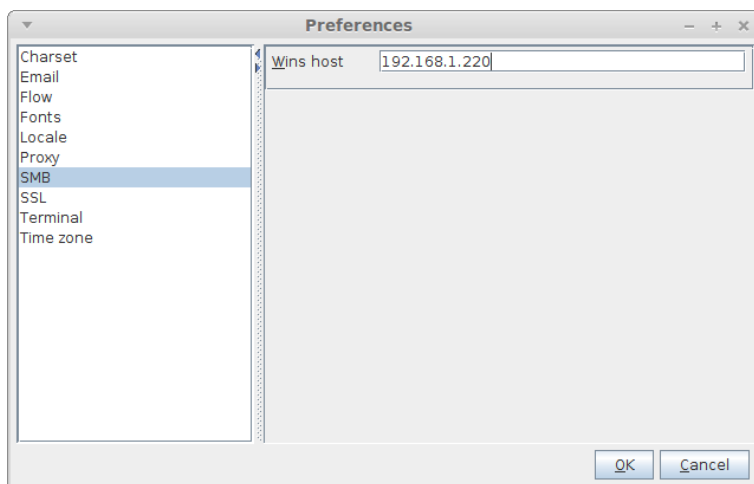


Figure 5.1: SMB preferences

The following actors are available:

- *SMBConnection* – standalone actor that defines the server connection (domain/user/password).
- *SMBGet* – for obtaining a remote file from a Windows host.
- *SMBSend* – for sending a file to a remote Windows host.

There are also several *searchlets* available for the *FileSystemSearch* source actor that allow the listing of remote files.



## Chapter 6

# Remote commands

The *adams-net* module adds several connection schemes for the remote command framework:

- *FTPConnection* – sends commands as files via FTP.
- *ScpConnection* – sends commands as files using secure copy (*scp*).
- *SSHConnection* – offers SSH tunnelling, useful when remote servers are locked down.



## Chapter 7

# Miscellaneous

Some other basic, but useful actors are the following:

- *Browser* – opens the system’s default browser with the specified URL.
- *DownloadFile* – downloads a file via HTTP and saves it to disk.
- *DownloadContent* – downloads (textual) content via HTTP and forwards it.<sup>1</sup>
- *HTMLFileReader* – reads an HTML file, generating a DOM object.
- *HttpRequest* – the source allows accessing URLs (POST/GET, sending data, cookies), returns the HTML from the reponse as text; the transformer allows sending the incoming text via HTTP request (and optional headers).
- *MimeType* – determines the mime-type of files.<sup>2</sup>
- *Socket* – source and sink allow you to receive and send strings and byte arrays via sockets.
- *URLSupplier* – outputs one or more URLs.
- *WebServer* – simple web server for serving static files.
- *WebSocketClient* – allows executing of clients that implement the websocket protocol<sup>3</sup>.
- *WebSocketServer* – allows running of websocket servers.

The following conversions are available:

- *HTMLToDOM* – generates a DOM document from an HTML string.
- *HTMLToText* – extracts the text from an HTML string.
- *RenderHTMLAsText* – renders an HTML string as text.
- *SpreadSheetToHTML* – converts a spreadsheet into an HTML page (or just a table fragment).
- *StringToURL* – converts a string into a URL object.
- *URLToString* – converts a URL object into a string.

In case you need to debug a connection, `telnet` is always a useful tool. ADAMS comes with a very simple graphical version, depicted in Figure 7.1.

---

<sup>1</sup>`adams-net-download_content.flow`

<sup>2</sup>`adams-net-mimetype.flow`

<sup>3</sup><https://en.wikipedia.org/wiki/WebSocket>

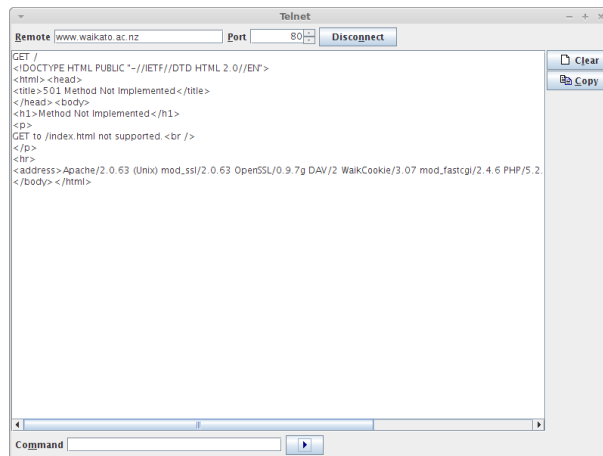


Figure 7.1: Telnet dialog

# Bibliography

- [1] *ADAMS* – Advanced Data mining and Machine learning System  
<https://adams.cms.waikato.ac.nz/>
- [2] *Gmail* – Configuring other mail clients  
<http://support.google.com/mail/bin/answer.py?hl=en&answer=13287>