

mmf 1.4

methodica message feeder

Administrator Guide

Document Code: mmf.7601.201008.en

Table of contents

Table of contents.....	1
1 Introduction.....	3
1.1 What it is	3
1.2 How it works.....	4
2 Prerequisites	10
2.1 Hardware	10
2.2 Software.....	10
3 Installation.....	12
3.1 Web Server and MySQL sources.....	12
3.2 MMF application installation.....	13
3.3 Dynamic loader installation	13
3.4 License file installation	14
4 Database and application setup.....	15
5 Administration	19
5.1 Tools and Utilities.....	19
5.2 The mmf/Backup utility.....	20
5.3 The mmf/MPump component.....	22
6 Configuration files	23
6.1 Overview	23
6.2 userpar.ini	24
6.3 _config.inc.....	25
6.4 _db.cfg	26
6.5 _debug.cfg	27
6.6 _default.cfg	27
7 Embedding a feed into a web page.....	29
7.1 The scripts	29
7.2 Tailoring the HTML code.....	30
7.3 The styles	30
8 Appendix	32
8.1 Scripts (JavaScript).....	32
8.2 Style sheets	34
8.3 Sample web page	36

Figures

Figure 1: System Architecture.....	4
Figure 2: Software Architecture.....	5
Figure 3: Dynamic Loader.....	6
Figure 4: Data Structure.....	6
Figure 5: Setup utility	15
Figure 6: Administration tools.....	19
Figure 7: Backup utility.....	20

Revision history

Date	Reason	Author
2010-08-10	First edition	hhe

Trademarks:

- UNIX is a registered trademark of The Open Group
- Linux is a registered trademark of Linus Torvalds
- Solaris is a registered trademark of Oracle Corporation
- Windows is a registered trademark of Microsoft Corporation
- methodica, methodica.ch, methodica message feeder, mmf/Editor, mmf/Setup, mmf/Backup, mmf/Report, and mmf/Sysinfo are trademarks of methodica herger, Switzerland.

Copyright:

We reserve all rights of authorship and ownership for this document and the software described herein.

Copyright © 2010, methodica herger.

1 Introduction

1.1 *What it is*

Methodica Message Feeder (MMF) is an application that supports...

- **publishing message feeds in these formats: RSS, ATOM, HTML,**
- **creating and editing the feed and message data.**

The data is stored in a database (currently MySQL).

The application includes components for...

- **providing the data to readers and browsers,**
- **editing feeds and messages,** and
- **importing messages from e-mail boxes.**

All components are working web-based. Thus, you can consume the messages and edit feeds and messages...

- **without the need to install a client component**
(standard browsers, such as Internet Explorer, Firefox, Chrome, etc., or RSS / ATOM reader applications will be sufficient),
- **from anywhere in the world or within an enterprise environment**
(internet / intranet access is necessary),
- **concurrently**
(many users may consume and edit the centrally stored feed data at the same time).

New feeds can be edited through an online editor interface.

Messages can be added through...

- the online editor interface, or
- the e-mail importer.

MMF currently is available in English and German.

Its components as well as the database support UTF-8 character sets. So, the feed and message data may include character sets like Japanese, Chinese, Russian, Greek, Thai, Hindi, Arabic, etc. MMF also supports the direction right-to-left (Arabic and Hebrew languages), although not all browsers and feed reader clients do so.

1.2 How it works

1.2.1 System Architecture

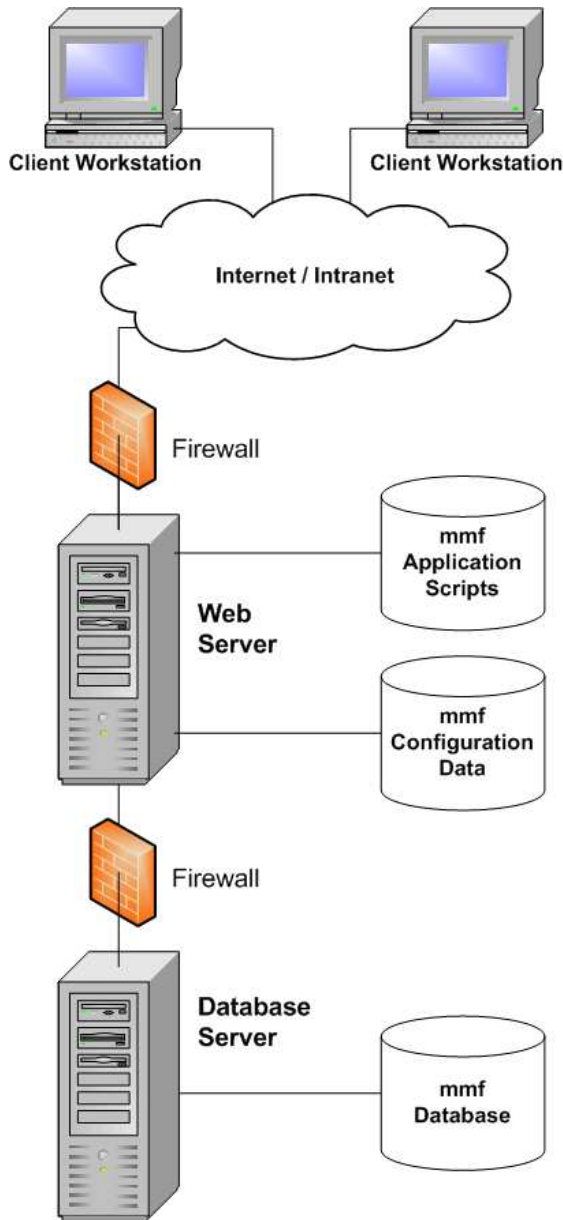


Figure 1: System Architecture

The application is integrated into a web server.

Besides the database service, all the components are web application scripts, written in PHP.

The web and database services may be installed on different machines or may reside on the same machine.

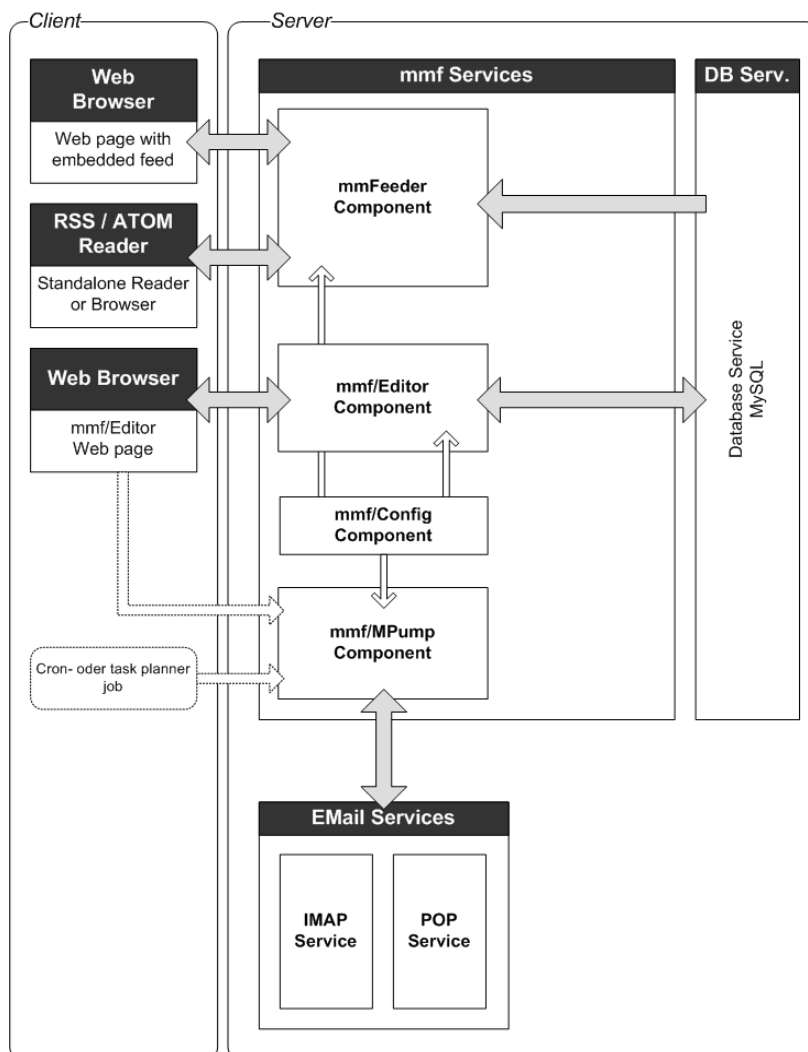
The browser installed on the client workstation, or an RSS / ATOM reader client application sends an HTTP request to the web server. The web server then includes the appropriate MMF component, which prepares the resulting text, and sends the result back to the client.

Feed data may be requested from within any web page (embedded, web-2 technology using Ajax¹ and JavaScript). The editor interface creates standard web pages (HTML).

For security reasons, in your environment, the web servers may be protected by firewalls, and you may be asked to authenticate before accessing the (intranet) web server and the application at all.

¹ AJAX = Aynchronous JavaScript and XML

1.2.2 Software Architecture



The MMF main components:

- **mmf/Feeder**
Provides feeds to consumers in all supported formats.
- **mmf/Editor**
Allows for online editing feed and message data in a comfortable way.
- **mmf/MPump**
Imports messages from e-mail accounts into the database. mmf/MPump accesses external e-mail services.

Figure 2: Software Architecture

On the client side, for all operations, a standard web browser is sufficient.
JavaScript must be available and active.

1.2.3 Dynamic Loader

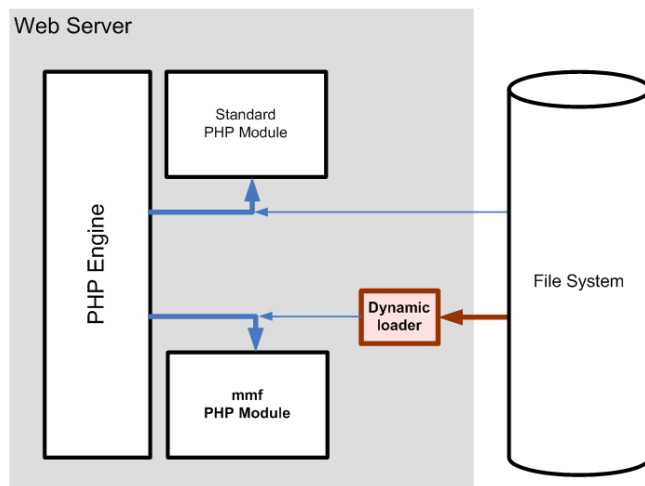


Figure 3: Dynamic Loader

The MMF modules are loaded using a dynamic loader module.

The dynamic loader...

1. decrypts the encrypted modules, and
2. checks them against the application license installed.

While the PHP engine reads standard (and also some MMF) modules directly from the file system the MMF modules are read by the dynamic loader module before they are executed.

1.2.4 Data

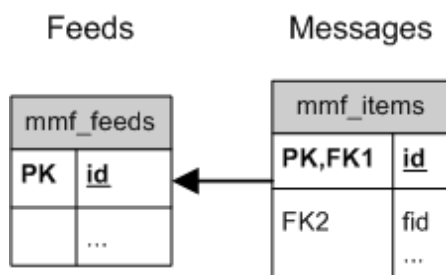


Figure 4: Data Structure

The mmFeeder data is stored in a database (currently MySQL).

The data is structured in just 2 entities:

- **Feeds**, and
- **Messages**.

Messages are assigned to feeds.

There may be any number of feeds, and any number of messages per feed.

Feeds are identified by their name, a 24-character string. Messages are linked to feeds via this name. Messages are identified by an automatically generated numeric key.

1.2.5 File Structure

The table below shows the application directories and files together with a brief description of its purposes.

The formal path name *path_to_mmfeed* means the drive (if appropriate) and path down to the MMF directory (e.g. *X:\webserver\www.mydomain.tld\mmfeed*).

Which files may be modified?

- Files that never may be modified are marked **red**,
- Those an administrator might modify carefully are marked **orange**, and
- Those an administrator may modify at his/hers will are marked **green**.

(See marks in the column titled **M**).

M	File	Purpose
path_to_mmfeed		
	_config.inc	Basic configurations
	_licinfo.php	Provides license information to be used by _sysinfo1.php (support only) and other components.
	_sysinfo1.php	Provides information about the application (for support only).
	_sysinfo2.php	Provides information about the server (for support only).
	feed_sample.html	Samole web page with embedded feed.
	index.php	mmf/Feeder component, logic.
	mmfeed_1_4.lic	License file.
	userpar.ini	User variables for reference in feeds and messages ² .
path_to_mmfeed/admin		
	_credentials.cfg	Username and password when the <i>admin</i> sub-directory is protected by .htaccess.
	_db.cfg	Database configuration ²
	_debug.cfg	Debug configuration ²
	_default.cfg	Default feed coniguration ² .
	admin_tpl.htm	mmf/Editor template for the user interface.
	index.php	mmf/Editor component, logic
	de.www.mydomain.tld.cfg	Feed configuration file , default, by setup
	en.www.mydomain.tld.cfg	Feed configuration file , default, by setup
	mail_feed.php	mmf/MPump component, logic
path_to_mmfeed/admin/archive		
	Database backup files.	
path_to_mmfeed/admin/doc		
	helpdoc.inc	Online help documentation structure.
	Miscellaneous help documentation files, language-specific.	
path_to_mmfeed/admin/images		
	Pictures and icons for use by mmf/Editor, language-specific.	
path_to_mmfeed/images		
	Pictures and icons for use by mmf/Feeder, language-specific.	
path_to_mmfeed/library		

² See 6 "Configuration files", page 23

M	File	Purpose
	langstrings.inc	Internationalisation functions
	MMF.class.php	mmf/Feeder, logic.
	MMF_editor.class.php	mmf/Editor, logic.
	MMF_message_form.inc	mmf/Editor, template and logic for the help and message windows.
	MMF_feed_demo.inc	mmf/Editor, template and logic for the sample page generator.
	MMF_feed_form.inc	mmf/Editor, template and logic feed form.
	MMF_feed_test.inc	mmf/Editor, logic feed test (mmf/Feeder emulation).
	MMF_form_functions.inc	mmf/Editor, common functions for handling forms.
	MMF_help_form.inc	mmf/Editor, template and logic for the help pages.
	MMF_item_form.inc	mmf/Editor, template and logic message form.
	MMF_login_form.inc	mmf/Editor, template and logic log-in form.
	MMF_vchk_form.inc	mmf/Editor, template and logic version checker.
	MMF_xmlreader.class.php	XML parser, logic.
pfad_zu_mmfeed/library/lang		
	de.txt	Import logic for text definitions, German
	import_lang.inc	Import logic for text definitions, general
	de_utf8.cod	Text definitions, German (Unicode)
	en-GB.txt	Import logic for text definitions, Englisch
pfad_zu_mmfeed/scripts		
	ajaxify.js ajaxify_min.js ³	Script for dynamically getting data from the server. Use for embedding feeds in your own web pages.
	jquery-1.4.2.min.js	mmf/Editor user interface, client logic.
	jquery-1.8.2.mmf.min.js	mmf/Editor user interface, client logic.
pfad_zu_mmfeed/setup		
	backup.php	mmf/Setup backup / restore module.
	createdb.inc	mmf/Setup database creation definitions.
	gentheme.php	mmf/Setup style sheet generator module.
	index.php	mmf/Setup main (overview) module.
	MMF_setup.class.php	mmf/Setup functions.
	setup.php	mmf/Setup database and application setup module.
pfad_zu_mmfeed/setup/lang		

³ compressed variant

M	File	Purpose
	Language definition files for set-up modules.	
	pfad_zu_mmfeed/styles	
	_default.css	Default styles (used when mmf/Feeder is directly called in HTML mode)
	mmfeed_blue.css	Feed class style definitions ⁴ , blue colour theme.
	mmfeed_default.css	Feed class style definitions ⁴ , default colour theme (usually equal to the orange colour theme).
	mmfeed_gray.css	Feed class style definitions ⁴ , gray colour theme.
	mmfeed_green.css	Feed class style definitions ⁴ , green colour theme.
	mmfeed_orange.css	Feed class style definitions ⁴ , orange colour theme.
	pfad_zu_mmfeed/styles/mmf	
	Miscellaneous styles and images for use with the mmf/Editor component.	
	pfad_zu_mmfeed/styles/themes	
	Miscellaneous style sheet and colour definitions.	

⁴ For embedding feeds in your own web pages. Also used by the mmf/Editor component together with the feed testing feature.

2 Prerequisites

2.1 Hardware

2.1.1 Architecture

On the server side, the application has a 2-tier layout with the presentation and the business logic on one machine and (optionally) the database on a second machine.

So, one or to machines suited to run a web server together with the MMF components, a database engine (MySQL), or both are needed.

For details see 1.2.1 "System Architecture", page 4.

2.1.2 Performance

The performance requirements can not be generally defined as it vastly depends on...

- the intensity the application will be used,
- the number of concurrent users, and
- co-existing applications on the same machine(s).

Generally speaking, the MMF application and the database service are well with standard server hardware.

A standard PC, however, can be considered as the minimum hardware required.

Some customers are running MMF together with a web site, with an Apache web server, the MMF application modules, and the MySQL database on one PC (Intel Pentium 4 CPU - 2.8 GHz, 3 GBytes of RAM, 80 GBytes hard disk space, Microsoft Windows Pro).

2.2 Software

Purpose	Software	Version	Remarks
Operating System	Unix ⁵ Linux ⁶ Solaris ⁷ Windows ⁸	n/a	• The application might run on other OS too, but has not been tested. • TCP stack for the web server and database must be available – of course.
Web Server	IIS Apache	5.0+ 2.0+, 1.3	• Other web servers might work too, but were not tested.
Database Server	MySQL, Community edition	5.0+	• The database server is connected using a TCP/IP socket or a Unix socket. Hostname / IP and port can

⁵ UNIX is a registered trademark of The Open Group

⁶ Linux is a registered trademark of Linus Torvalds

⁷ Solaris is a registered trademark of Oracle Corporation

⁸ Windows is a registered trademark of Microsoft Corporation

Purpose	Software	Version	Remarks
			be configured.
Scripting Engine	PHP engine	4.3+	• Dynamic loading (DL) must be enabled, i.e. ... - o in php.ini: enable_dl=on, - o PHP not in safe mode, - o PHP not compiled with thread safety (e.g. on Windows when PHP is installed as a module). • Access to the extension directory is necessary with some versions. • IMAP / POP support is necessary for the mail import feature (mmf/MPump).

3 Installation

To have MMF running, these steps are necessary:

1. **(If not yet done) Install a web server, PHP, and the MySQL database.**
2. **Get the application archive file, expand it and copy it into a sub-directory of the web server root directory.**
3. **Get the loader module necessary to load the MMF and install it.**
4. **Get a license and copy the license file into the MMF base directory.**
5. **Perform the setup process to create the database and to configure the application.**

We do not describe in detail how to install and set up a web server and database infrastructure. Most customers will already have one in place.

On Linux / Unix operating systems, an Apache web server and the MySQL database are either pre-installed or can easily be installed using the built-in installers.

On Windows server operating systems you usually find an Internet Information Server (IIS) pre-installed. Instead of the MySQL database often SQL Server from Microsoft is installed.

MMF currently only supports MySQL for the simple reasons that...

- up to now customers never asked for SQL Server or Oracle database support⁹, and
- MySQL is performing well, is stable, and has a zero price tag.

3.1 Web Server and MySQL sources

For your convenience, we provide some links from where to download the software necessary to meet the prerequisites.

Note: These links were valid at the time this document was edited. If you can not find the appropriate software please use a search engine to find the sources.

All-in one: XAMPP

- for Linux: <http://www.apachefriends.org/en/xampp-linux.html>
- for Windows: <http://www.apachefriends.org/en/xampp-windows.html>

The packages you download from here embrace the Apache web server, the MySQL community edition database engine, PHP, an FTP server, a control panel for all the main components and a couple of utilities and tools you can optionally install too.

All these components are freeware.

⁹ We will consider supporting other database engines too if we see a need for it.

We recommend you install XAMPP if you do not yet have a web server and database infrastructure in place.

If you, however, prefer to install single components see below.

Web server

- for Linux / Windows: <http://httpd.apache.org/download.cgi>

You can download the source code for both operating systems and compile it yourself. Pre-compiled binaries are available for the Windows operating system.

PHP

- for Linux / Windows: <http://php.net/downloads.php>
- Binaries for Windows: <http://windows.php.net/download/>

Database

- for Linux / Windows: <http://dev.mysql.com/downloads/mysql/>

You will find binaries for several operating systems there, and you also can download the source code and compile it yourself.

3.2 MMF application installation

Download the MMF application archive appropriate for your operating system:

http://www.methodica.ch/hml_en/tools/mmfeeder/downloads/

You can transfer it to the web server machine and expand it there, or you can expand it on your client workstation and copy the contents of the resulting directory and all its subdirectories to the web server. For copying MMF to your web server, you obviously need the access credentials and an FTP client software.

Since MMF is a web application it must be stored in a sub-directory of the directory configured as the web site's root directory, e.g...

- **`/var/www/www.yourdomain.tld/Web/mmfeed`**¹⁰, or
- **`F:\web\www.yourdomain.tld\mmfeed`**¹⁰

This directory is called the MMF base directory. We recommend you name it *mmfeed*, but you are free to choose any other name.

The MMF web application is then called through URLs like these:

- **`http://www.yourdomain.tld/mmfeed`**¹⁰ (component *mmf/Feeder*),
- **`http://www.yourdomain.tld/mmfeed/admin`**¹⁰ (component *mmf/Editor*),
- **`http://www.yourdomain.tld/mmfeed/setup`**¹⁰ (component *mmf/Setup*).

3.3 Dynamic loader installation

Most MMF components are loaded through a dynamic loader module. Usually you do not have it installed already.

¹⁰ Replace *www.yourdomain.tld/mmfeed* by the correct MMF base path name.

First, these requirements must be fulfilled for the loader module to properly work:

- In PHP.INI: set *enable_dl=on*,
- PHP must not be executed in safe mode as this inhibits dynamic loading of modules at all,
- PHP must not be compiled with thread safety (e.g. on Windows when PHP is installed as a module).

Secondly, download and copy the loader module that corresponds to the operating system from here: <http://www.sourceguardian.com/ixeds/>, and copy it to the appropriate directory. The directory depends on the operation system and PHP version installed.

You can find instructions where to install which loader module with the help of a utility delivered together with the MMF application. In a web browser enter an URL like this: ***http://www.yourdomain.tld/mmfeed/_install_loaders.php***¹⁰ and follow the instructions.

3.4 License file installation

In order to run any of the MMF components (including mmf/Setup) a valid license file is necessary.

If you bought a license you will have received this license file by e-mail or have been instructed how to download it.

Copy this file, named ***mmfeed_1_4.lic***, into the MMF base directory, e.g. */var/www/www.yourdomain.tld/Web/mmfeed/mmfeed_1_4.lic*.

You are welcome to ask for an evaluation license, valid for 30 days (or more if you need it). With evaluation licenses, MMF has the same functionality as with other licenses.

All licenses (including evaluation licenses) are requested through this URL: ***http://www.methodica.ch/hml_en/tools/mmfeeder/licenses/***.

4 Database and application setup

After you have installed the application, the dynamic loader module, and the license file you are ready to set up the database and the application.

In a web browser enter this URL:

http://www.yourdomain.tld/mmfeed/setup/setup.php¹⁰, or

http://www.yourdomain.tld/mmfeed/setup¹⁰, and then click the *mmf/Setup* link.

A web page with a form like the one shown below will be displayed.

methodica message feed setup utility 1.4.0

Copyright © 2010, methodica.ch

Status: Ready

Database Configuration

Group	Parameter	Value	Status
Database Host	Hostname / IP	localhost	2 - Not set up
	Port	3306	
Root User	Name		
	Password		
MMFeed User	Name	mmf_tu	2 - Not set up
	Password	mmf4C920BC5227C63@23768979	
Database	Name	mmfeed	
	Table Name Prefix	mmf_	

System Configuration

Group	Parameter	Value	Status
File structure	HTTP path:	http://www.methodica.ch/mmfeed1/	0 - OK
	Root path:	/var/www/www.methodica.ch/Web/mmfeed1/	
	Administration sub-directory:	admin/	
	Images sub-directory:	images/	
	Library sub-directory:	library/	
	Languages sub-directory:	library/lang/	
	Scripts sub-directory:	scripts/	
Defaults	Styles sub-directory:	styles/	2 - Not set up
	Language Code	English (Great Britain) ▼	
	Time-to-live	60 Minutes	
Debugging	Publication time	1 Weeks ▼	2 - Not set up
	Debug file name	/var/www/www.methodica.ch/Web/mmfeed1/admin/debug.txt	
	Debug file size	300 KBytes ▼	
	Active	☐ On / Off	
	Verbose	☐ On / Off	

Figure 5: Setup utility

The fields are grouped:

- **Database Configuration**
(to create the database, the database tables, and to populate them with demo contents), and

- *System Configuration*
(all other parameters).

Parameters that tightly belong to each other are further grouped into sub-groups.

A status marker per sub-group shows its status:

- Not set up: These parameters are not configured yet or the database has not been set up yet.
- Error: One or more of the parameters are mandatory and have no content or have formally illegal contents.
- OK: The parameters are ok and the corresponding set-up process was successful.

The status markers are updated every time you click one of the *Setup Configuration* or *Refresh Configuration* buttons.

Fill in the fields (details see below). When done, click the *Setup Configuration* button. If only a part of the set-up process have been successful you still can modify the parameters and try again.

The fields in detail:

Field label	Purpose	Remarks
Database Configuration		
Database Host		
Hostname / IP	Hostname or IP of the database server.	Fully qualified host name, e.g.: mysql.mydomain.tld. Mandatory. Default: <i>localhost</i>
Port	Port number of the database service.	Mandatory. Default: <i>3306</i>
Root User	User with privilege to create a new database (at least to create and delete tables).	
Name	-	
Password	-	
MMfeed User	Technical user the MMF components will use to connect to the database. If not present before, this user will be granted limited rights.	
Name	-	
Password	-	
Database		
Name	Database name	e.g.: mmfeed. Mandatory.
Table name prefix	Table name prefix	e.g.: mmf_ Optional,

Field label	Purpose	Remarks
		strongly recommended when using an existing database.
System Configuration		
File structure	Read-only. The setup utility shows the directories found within the base directory.	
Defaults	Default values to use when creating a new feed. May be overwritten by feed authors.	
Language Code	Feed language and language default language of mmf/Editor.	Choose one from the list of installed languages. Default: <i>English (Great Britain)</i> .
Time-to-live	Time in minutes a feed consumer may locally cache a feed before it should be re-read.	Mandatory. Default: 60 (1 hour).
Publication time	Time frame a message shall be published, beginning with the publishing date.	Mandatory. Default: 1 week.
Debugging	Debug parameters to use by the MMF components. You might be asked to activate debugging by the manufacturer's support personnel.	
Debug file name	Path and name of the debug file.	Mandatory. Default: (by mmf/Setup).
Debug file size	Maximum size of the debug file. When this size is reached a new debug file will be created.	Optional. Default: 300Kbytes. Unlimited if this value is 0 (zero).
Active	Activate / inactivate debugging.	Check to activate. Default: inactive.
Verbose	Activate / inactivate verbose debugging mode.	Check to activate. Default: inactive.

Once you have properly set up the application, for security reasons, we recommend you rename the setup directory and manually adjust the corresponding value in the base configuration file `_config.inc`.

E.g. rename the directory from...

`/var/www/www.yourdomain.tld/Web/mmfeed/setup/` to ...
`/var/www/www.yourdomain.tld/Web/mmfeed/admin_brown/`

Then change the appropriate value in the file ...
/var/www/www.yourdomain.tld/Web/mmfeed/_config.inc:

```
<?php
/*
-----
MMFeeder Configuration.
File:      /var/www/www.methodica.ch/Web/mmfeed/_config.inc
Created: 2010-09-16 15:06:14 (UTC) by Setup.
-----
*/
...
define ( 'SETUPPATH', 'admin_brown/' );
...
?>
```

Additionally / alternatively, you might protect access to the setup directory with a password using the *.htaccess* mechanism. For more information about this mechanism see here: <http://httpd.apache.org/docs/current/howto/htaccess.html>.

5 Administration

A tool set helps you, the administrator, to maintain the MMF application system.
In a web browser enter this URL:

<http://www.yourdomain.tld/mmfeed/setup/>^{10,11}

mmf/Admin 1.04.00		
methodica message feed administrator utilities Copyright © 2010, methodica.ch		
Administrator Utilities		
Name	Purpose	Link
mmf/Setup	Set up the system: • Create the database if it not yet exists. • Create the database tables. • Create a (technical) database user. • Add sample feed and message data. • Write a database configuration file for further use by the mmFeeder components. • Write a debug configuration parameter file. • Write other configuration parameters to a file.	
mmf/Backup	Back up or restore the database • Show the list of backup files. • View the contents of a backup file. • Save the database contents to a backup file. • Restore the database from a backup file.	
mmf/Report	Installation report Show installation and licence information. Also necessary for remote support. (In English only)	
mmf/Sysinfo	System information Show system information (phpinfo). Also necessary for remote support. (In English only)	
mmf/Editor	Feed Editor Online feed and message online editor • Create / edit / delete feeds and messages. • Test feeds together with its messages. • Generate sample web page code.	

Figure 6: Administration tools

5.1 Tools and Utilities

These tools and utilities are available:

- **mmf/Setup**
The utility is described in chapter 4 "Database and application setup" on page 15.
- **mmf/Backup**
A utility to take backup snapshots from the database or to restore the database from a snapshot file.

¹¹ instead of *setup*, use the correct name if you renamed the directory.

It is a good idea to take single snapshots before you make fundamental changes to the feed structure. Regularly take snapshots (e.g. on a daily basis) to prevent loss of data when the disk holding the database gets corrupt or when a feed editor unwillingly deletes messages or feeds.

The backup utility will be described in more detail below (see 5.2 "The mmf/Backup utility", page 20).

- **mmf/Report**
This tool creates a report showing the installation and license information. It is one of two reporting tools necessary whenever you need support from the manufacturer. It is web-based as all other tools and utilities are, so, it is available to the manufacturer's support personnel if you grant them access.
- **mmf/Sysinfo**
This tool provides the same information than the PHP function "phpinfo" does. It covers the relevant information about the PHP engine installed on your web server.
- **mmf/Editor**
This is the tool every feed author / editor needs. It is described in the separate [user manual](#).
An administrator might use mmf/Editor for testing the installation and when an author / editor forgot the username and / or password used to protect a feed. In this case, you could authenticate using the MMF User name and password pair you specified during setting up the application (see chapter 4 "Database and application setup", page 15, description of "MMfeed User") and set new username and password values.
- **mmf/MPump**
This utility imports messages from specially formatted e-mails and stores them in the database. mmf/MPump is automatically started by the mmf/Editor component and also can be called from batch jobs or via the console.

5.2 The mmf/Backup utility

mmf/Backup 1.04.00
methodica message feed backup utility
Copyright © 2010, methodica.ch

Backup / Restore mmFeeder database

Click an icon from the list below (*) to make a backup copy, to restore the database from a backup copy, or to remove a backup copy.

File	Date / Time	*	Remove
Make a backup from the current database			
mmf_backup.xml	mmf_backup.xml		
db_20100910_163725.xml	2010-09-10 16:37:25		
db_20100910_094641.xml	2010-09-10 09:46:41		
Refresh the above list			

Back-up the database

Figure 7: Backup utility

You can take a snapshot from the current database contents by clicking the corresponding (green arrow) icon on the top of the list of snapshot files, in the column titled "*".

The backup utility web page shows the list of the snapshot files from back-ups taken earlier. For every file you can...

- Restore the database to the contents it had when the snapshot was taken. To restore click on the appropriate (red arrow) icon in the column titled "*".
Note: if possible, we recommend you take a new backup snapshot before restoring the database from a snapshot file. So, in case it did not contain what you were looking for, you could recover the current state.
- View the snapshot contents (XML formatted).
Clicking on the name of the file will display its contents online.
- Remove the snapshot file.
When you click the (red cross) icon in the column titled "Remove" the corresponding snapshot file will be deleted.
Note: this operation can not be undone.

Finally, you can refresh the list of snapshot files by clicking the corresponding (grey document with arrows) icon in the column titled "*" below the list.

The backup utility is available in English only.

Command line interface

The backup utility script can also be called from a batch job, e.g. started periodically by Cron (Unix / Linux) or Task Scheduler (Windows).

To do so, use the PHP command line interface (PHP CLI) like this:

- for Linux:
`/usr/bin/php 12 /var/www/www.yourdomain.tld/Web/mmfeed/setup/backup.php 10 options`
- for Windows:
`C:\prog\php.exe 12 "X:\webserver\www.mydomain.tld\mmfeed\setup\backup.php" 10 options`

Options depend on the desired operation:

- **Backup:**
`c=b`
- **Restore:**
`c=r&f=path_and_name_of_snapshot_file`
e.g.:
`c=r "f=X:\webserver\www.mydomain.tld\mmfeed\setup\admin\my archive\db_20100910_094641.xml"`
- **Remove (delete):**
`c=d&f=path_and_name_of_snapshot_file`
e.g.:
`"c=d&f=/var/www/www.yourdomain.tld/Web/mmfeed/admin/my archive/db_20100910_094641.xml"`

¹² Set the correct path and file name of the PHP engine (CLI) executive here.

Formatting rules for options:

- Option parameters are keyword parameters of the form *key=value*.
- Option parameters are separated from each other by either the ampersand (&) character or the blank space character ().
- With the blank space separator character, if a parameter contains a blank space embrace it with double quote characters ("...").
With the ampersand separator character, if a parameter contains a blank space embrace the entire options string with double quote characters.

Examples of batch jobs can be found here...

- for Linux:
path_to_mmfeed/setup/**backup_nix.bat**
- for Windows:
path_to_mmfeed/setup/**backup_win.bat**

5.3 The mmf/MPump component

When supported by the PHP version messages may be added via specially formatted e-mails. The component that scans mail boxes and imports messages from e-mails is called mmf/MPump.

This component is automatically called when someone uses mmf/Editor, the online editor component. mmf/Editor periodically issues a call to mmf/MPump as long as the editor web page is open. This feature may be de-activated by setting the parameter *SUPPRESS_MAILPUMP* in the configuration file path_to_mmfeed/_**config.inc** (see 6.3 "_config.inc", page 25) to *true*.

The script path_to_mmfeed/admin/**mail_feed.php** is the implementation of the mmf/MPump component. It can also be called via command line interface, e.g. from a batch job started by Cron (Unix / Linux) or Task Scheduler (Windows).

Command line interface

To call mmf/MPump, use the PHP command line interface (PHP CLI) like this:

- for Linux:
/usr/bin/php¹² /var/www/www.yourdomain.tld/Web/mmfeed/admin/**mail_feed.php**¹⁰ option
- for Windows:
C:\prog\php.exe¹² "X:\webserver\www.mydomain.tld\mmfeed\admin**mail_feed.php**"¹⁰ option

Option is optional. When called with the option **verbose=true** mmf/MPump will display information about the import process.

Examples of batch jobs can be found here...

- for Linux:
path_to_mmfeed/admin/**mailfeed_nix.bat**
- for Windows:
path_to_mmfeed/admin/**mailfeed_win.bat**

6 Configuration files

6.1 Overview

The table below lists the configuration files and briefly describes what they define. Parameters that should not be modified are marked **orange**, those you can modify are marked **green**.

File		
M	Parameter	Purpose, default value
path_to_mmfeed/ userpar.ini		
	Company	Company name
	(You can add your own variables which can be referenced in feed and message fields)	
path_to_mmfeed/_ config.inc		
	SUPPRESS_VERSIONCHECK	mmf/Editor does not provide a version check link.
	SUPPRESS_MAILPUMP	mmf/Editor does not start mmf/MPump.
	OPEN_MESSAGES	In HTML format, mmf/Feeder shows this number of messages expanded and the rest compacted. Set this parameter to -1 if no messages should be compacted. Default: 3.
	(All other parameters are defined when the product has been set up)	
path_to_mmfeed/admin/_ debug.cfg		
	DEBUGFILE	Path and name of the debug file
	DEBUG	Debug mode <i>true false</i> , default: <i>false</i>
	DEBUGVERBOSE	Verbose debugging <i>true false</i> , default: <i>false</i>
	DEBUGMAXSIZE	Maximum size of the debug file. If this value is reached a new debug file will be created. Default: 307200 (= 3 KB)
path_to_mmfeed/admin/_ default.cfg		
	LANG	Language code for mmf /Editor and default language code for feeds (the language must be installed). Default: <i>en-GB</i> .
	TTL	Time-to-live: defines for how many minutes a feed reader should locally cache a

File		
M	Parameter	Purpose, default value
		feed before requesting it again. Default: 60 minutes.
	MAXAGE	Default time frame length <i>Publication time from...until</i> . Default: 604800 seconds (= 7 days).
path_to_mmfeed/admin/_credentials.cfg		
Note: Only necessary if access to the <i>admin</i> directory is protected by .htaccess.		
	username	Username, default: none
	password	Password, default: none

Notes: Do not modify other configuration files not listed here.
Carefully follow the syntax of the modification files - otherwise the application might become unusable!

6.2 userpar.ini

Variables defined here may be referenced in most of the contents of feed and message fields by an author / editor. Variable references are variable names embraces in curly brackets.

So, for example one could set the contents of the field "Copyright" to "Copyright {copy} {year}, {company}".

Contents of userpar.ini

```
; Entry format: "key=value".
; Comment lines start with semikolon
; company=myCompany
; author=FeedMaster
```

There are 2 variables defined in this file – although commented out (;): *company* and *author*.

These standard variables are always available:

Variable name	Meaning	Remarks
{server_root_url}	URL to the home page of the web site.	» http://www.mydomain.tld
{date}	Date in the format "yyyy-mm-dd".	» 2013-05-21
{time}	Time in the format "hh:mm:ss", 24 hours.	» 22:03:27
{year}	Year in the format "yyyy"	» 2011
{copy}	The copyright sign © in a format suited for feeds.	
{httpbasepath}	URL pointing to the feed root (not the home page).	» https://www.mydomain.tld/mmfeed/
{filebasepath}	File path of the feed directory.	For use with your own scripts.
{your_variable}	Additionally, every variable that is defined in the file: path_to_mmfeeder/userpar.ini may be referred to in the same way.	

6.3 `_config.inc`

This is the basic configuration file for all MMF components. It is created during the application set-up process.

Contents of `_config.inc`

```
<?php
/*
-----
MMFeeder Configuration.
File:      /var/www/www.methodica.ch/Web/mmf/_config.inc
Created: 2010-09-13 08:20:38 (UTC) by Setup.
-----
*/
define ('LANGUAGES','English (Great Britain):en-GB|Deutsch:de');
define ('HTTPBASEPATH','http://www.methodica.ch/mmf/');
define ('FILEBASEPATH','/var/www/www.methodica.ch/Web/mmf/');
define ('FEEDDIR','mmf');
define ('CLASSPATH','library/');
define ('LANGPATH','library/lang/');
define ('ADMPATH','admin/');
define ('DOCPATH','doc/');
define ('IMAGESPATH','images/');
define ('SCRIPTSPATH','scripts/');
define ('STYLESPATH','styles/');
define ('SETUPPATH','setup/');
define ('CFGEXT','.cfg');
define ('STYLESEXT','.css');
define ('DEFAULTCFG','_default.cfg');
define ('DBCFG','_db.cfg');
define ('DEBUGCFG','_debug.cfg');
define ('DEFSTYLES','_default.css');
define ('LINKIMAGE','link_arrow.png');
define ('SUPPRESS_MAILPUMP',false);
define ('SUPPRESS_VERSIONCHECK',false);
define ('OPEN_MESSAGES',3);
?>
```

The entries:

M	Definition	Meaning	Remarks
	LANGUAGES	The languages installed.	
	HTTPBASEPATH	MMF path for use in URLs.	
	FILEBASEPATH	File system MMF path.	
	FEEDDIR	MMF directory (below host root).	
	CLASSPATH	Classes and functions directory.	
	LANGPATH	Languages directory (below CLASSPATH).	
	ADMPATH	mmf/Editor feed and messages admin. path.	
	DOCPATH	Documents and help directory (below ADMPATH).	
	IMAGESPATH	Images directory.	
	SCRIPTSPATH	Client-side scripts directory.	
	STYLESPATH	Style sheets directory.	
	SETUPPATH	Setup / Administrator directory.	
	CFGEXT	Standard extension of configuration files.	

M	Definition	Meaning	Remarks
	STYLESEXT	Standard extension of style sheet files.	
	DEFAULTCFG	Name of default configuration file.	
	DBCFG	Name of database configuration file.	
	DEBUGCFG	Name of debugging configuration file.	
	DEFSTYLES	Default style sheet if none is specified.	
	LINKIMAGE	Link arrow to use in HTML formatted feeds.	
	SUPPRESS_MAILPUMP	De-activates the automatic e-mail importer.	<i>true or false</i>
	SUPPRESS_VERSIONCHECK	De-activates the version check feature.	<i>true or false</i>
	OPEN_MESSAGES	Number of messages to show expanded.	Default: 3

The meaning of the colour markers (M):

: Do not modify at all.

: Modify only when you re-name the setup directory for security reasons.

: Modify, but be careful to follow the syntax.

6.4 _db.cfg

This file configures the database access. It is created during the application setup process. **Do not modify anything here!**

Contents of _db.cfg

```
<?php
/*
-----
MMFeeder Database Configuration.
File:      /var/www/www.mydomain.tld/Web/mmf/admin/_db.cfg
Created: 2010-09-02 10:31:20 (UTC) by Setup.
-----
*/
define ('DB_PREF', 'mmf_');
define ('DB_HOST', 'localhost');
define ('DB_PORT', '3306');
define ('DB_NAME', 'mmfeed');
define ('DB_FTAB', 'mmf_feeds');
define ('DB_ITAB', 'mmf_items');
define ('DB_USER', 'mmf_tu');
define ('DB_PWOR', 'mmf4C935CCE9696A9@41800409');
?>
```

The entries:

Definition	Meaning	Default
DB_PREF	Table name prefix.	mmf_
DB_HOST	Database service host name or IP	localhost
DB_PORT	Database service port number	3306
DB_NAME	Database name	mmfeed
DB_FTAB	Name of the feeds table	mmf_feeds
DB_ITAB	Name of the messages table	mmf_items
DB_USER	Name of the MMF database user	mmf_tu
DB_PWOR	Password for the MMF database user	(generated)

6.5 *_debug.cfg*

This file configures debugging. You may change these values if you honour the syntax. Best, you do it using the mmf/Setup component.

Contents of *_debug.cfg*

```
<?php
/*
-----
MMFeeder Debug Configuration.
File:      F:/www.mydomain.tld/mmfeed/admin/_debug.cfg
Created:   2010-09-19 13:56:39 (UTC) by Setup.
-----
*/
define ('DEBUGFILE', 'W:/www.mydomain/mmfeed/admin/debug.txt');
define ('DEBUG', true);
define ('DEBUGVERBOSE', false);
define ('DEBUGMAXSIZE', 307200);
?>
```

The entries:

Definition	Meaning	Default
DEBUGFILE	Debug file path and name.	n/a
DEBUG	Debugging on (true) or off (false)	false
DEBUGVERBOSE	Verbose debugging on (true) or off (false)	false
DEBUGMAXSIZE	Maximum debug file size. A new file is created when this value is reached.	300Kbytes (307200B)

6.6 *_default.cfg*

This file configures default values for feeds. All parameters may be overwritten when you create or edit a feed. The parameter *LANG* is also used for the components such as mmf/Editor, mmf/Setup, etc. You may change these values if you honour the syntax. Best, you do it using the mmf/Setup component.

Contents of *_default.cfg*

```
<?php
/*
-----
MMFeeder Defaults Configuration.
File:      F:/www.mydomain.tld/mmfeed/admin/_default.cfg
Created:   2010-09-19 13:56:39 (UTC) by Setup.
-----
*/
define ('LANG', 'en-GB');
define ('TTL', 60);
define ('MAXAGE', 604800);
$languages = array(
    'English (Great Britain)' => 'en-GB',
    'Deutsch' => 'de'
);
?>
```

The entries:

Definition	Meaning	Default
LANG	Default language to use with MMF components. Does not apply to feeds.	<i>En-GB</i>
TTL	Default time-to-live value (minutes). TTL is the maximum time a feed reader may locally cache a feed.	<i>60</i>
MAXAGE	Default difference between publication date / time from...until in seconds.	<i>1 week (604800s)</i>

7 Embedding a feed into a web page

Feeds (including messages) are published in RSS, ATOM, and HTML format.

You can embed a feed in HTML format into any web page.

Three steps are necessary to embed a feed:

1. The page must contain the scripts to request the feed and assign the resulting text to a container (DIV) within the page (feed update).
2. The HTML code must be tailored to include the container and to start the feed update script.
3. Styles are necessary to define how the elements of the feed are displayed.

The easiest way to create a working sample page which contains it all is to use the *Sample page (code)* feature of the mmf/Editor component (see the [user guide](#)).

7.1 The scripts

A web page with an embedded feed should contain this code in the `<head>...</head>` part:

```
<script type="text/javascript" src="http://www-  
mydomain.tld/mmfeed/scripts/ajaxify_min.js"></script>  
  
<script type="text/javascript">  
  var refreshInterval = 60 * 1000; // 60 seconds  
  var feedTimer = 0;  
  var ajax_obj = 0;  
  
  function loadFeed() {  
    if (feedTimer) {  
      clearTimeout(feedTimer);  
      feedTimer = 0;  
    }  
    ajax_obj = new ajax();  
    ajax_obj.openRequest(  
      'index.php',  
      'feedname=en.www.mydomain.tld&type=html&options=inline|nohead',  
      'GET',  
      'news_panel'  
    );  
    feedTimer = setTimeout('loadFeed()',refreshInterval);  
  }  
</script>
```

The first script (...*ajaxify_min.js*) includes a script that handles the AJAX communication and data transfer with the host. This is necessary to dynamically load parts of a web page. You can find a listing of this script here: 8.1.1 "ajaxify.js", page 32.

The second script - the feed update script - automatically and repeatedly (time interval: 60 seconds, may of course be changed) sends a request towards the

web server to get the feed. The resulting text is assigned to a DIV element in the web page whose id is `news_panel`.

The update loop is initialised by an initial call of the function `loadFeed()`.

Note: replace the formal feed name `en.www.mydomain.tld` with the real name of the feed you want to embed.

7.2 Tailoring the HTML code

Within the HTML code of a web page, a DIV container with the correct id must be available to take the feed contents the feed update script will assign to it. For example:

```
<div id="news_panel">&nbsp;</div>
```

Note: the id (`news_panel`) must be equal to the corresponding parameter in the feed update script.

And, the BODY tag of the page must include a call to start the feed update loop. For example:

```
<body onload="loadFeed();">
```

7.3 The styles

You can modify the look of an embedded feed by class styles. mmf/Feeder assigns a class to every part of a feed. Styles define how elements of a class are displayed.

This is the set of classes:

Class name	Element / State
<code>.mmf_feed_panel</code>	Feed panel (e.g. border, background).
<code>.mmf_feed_header</code>	Feed title.
<code>.mmf_feed_header_link</code>	Hyperlinks in the feed title (if appropriate).
<code>.mmf_feed_header_link:hover</code>	Same as above; when the mouse is over the element.
<code>.mmf_feed_text</code>	Feed text.
<code>.mmf_feed_text_link</code>	Hyperlinks in the feed text.
<code>.mmf_feed_text_link:hover</code>	Same as above; when the mouse is over the element.
<code>.mmf_feed_footer</code>	Feed footer (e.g. copyright text).
<code>.mmf_feed_footer_link</code>	Hyperlinks in the feed footer.
<code>.mmf_feed_footer_link:hover</code>	Same as above; when the mouse is over the element.
<code>.mmf_item_panel</code>	Message panel (e.g. border, background).
<code>.mmf_item_panel_last</code>	Last message panel.
<code>.mmf_item_header</code>	Message title.
<code>.mmf_item_header_link</code>	Hyperlinks in the message title (if appropriate).
<code>.mmf_item_header_link:hover</code>	Same as above; when the mouse is over the element.
<code>.mmf_item_text</code>	Message text.
<code>.mmf_item_text_link</code>	Hyperlinks in the message text.

Class name	Element / State
.mmf_item_text:hover	Same as above; when the mouse is over the element.
.mmf_item_footer	Message footer, e.g. publication date, author, etc.
.mmf_item_footer_link	Hyperlinks in the message footer.
.mmf_item_footer_link:hover	Same as above; when the mouse is over the element.
.mmf_feed_text_link IMG	Images in hyperlinks within the feed text.
.mmf_item_text_link IMG	Images in hyperlinks within the message text.
.mmf_p	Paragraphs within the feed and message text.

Style sheet files contain the style definitions for the classes. They are included in the <head>...</head> part of a page like this:

```
<link type="text/css"
href="http://www.yourdomain.tld/mmfeed/styles/mmfeed_default.css"
rel="stylesheet" />
```

A default style sheet file plus the variants *blue*, *gray*, *green*, and *orange* are pre-defined and can be found in the directory "path_to_mmfeed/styles/":

- *mmfeed_default.css* (the same as *mmfeed_orange.css*),
- *mmfeed_blue.css*,
- *mmfeed_gray.css*,
- *mmfeed_green.css*,
- *mmfeed_orange.css*.

You might wish to create your own style sheet in order for embedded feeds to match the styles of your web pages.

In this case we recommend you modify the necessary attributes based on one of these style sheet files, and save the resulting file under a name of the form "*mmfeed_scheme.css*"¹³ in the directory "path_to_mmfeed/styles/". This will allow you to test feeds by the mmf/Editor component using this style sheet, and to easily verify the style sheet.

¹³ Use your style sheet name instead of the formal name *scheme*.

8 Appendix

8.1 Scripts (JavaScript)

A couple of client-side scripts are necessary for the mmf/Editor component and for embedding a feed into a web page. These scripts are written in JavaScript, version 1.1.

8.1.1 ajaxify.js

This script must be included into web pages with embedded feeds. The script comes in a verbose and a compressed variant. The compressed variant loads faster in the browser so you might preferably include this script in your web pages.

You carefully could extend this script if there is need for, or include its code in a script of your own.

The original, verbose, variant of the script is shown below.

```
/* *****  
  
AJAXIFY.JS: A simple AJAX script  
  
Copyright (C) 2010 methodica (http://www.methodica.ch)  
  
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, version 3 of the License.  
  
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
The author may UNDER NO CIRCUMSTANCES BE MADE LIABLE FOR ANY  
CONSEQUENCES OF USING THE PROGRAM. IF THIS WAIVER OF ALL LIABILITY  
CONTRADICTS THE LAW WHERE YOU INTEND TO USE THE PROGRAM ITS USE IS  
STRICTLY PROHIBITED.  
See the GNU General Public License for more details.  
  
You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.  
  
***** */  
  
// Ajax state values  
var READY_STATE_UNINITIALIZED = 0;  
var READY_STATE_LOADING = 1;  
var READY_STATE_LOADED = 2;  
var READY_STATE_INTERACTIVE = 3;  
var READY_STATE_COMPLETE = 4;  
  
// Ajax request counter  
var OPEN_AJAX_REQUESTS = 0;  
  
// Get element object from element id  
function getElement(id) {  
    return document.getElementById ? document.getElementById(id) : document.all ? docu-  
ment.all(id) : document.layers[id];  
}  
  
// Expand / contract containers  
function togglepnl(id){  
    if(!id) { return; }  
    id = typeof id === "string" ? getElement(id) : id; if(!id) return;
```

```
    var expand = (id.style.display=='none');
    id.style.display=(expand ? "" : "none");
}

// Adjust element alignment and text direction for arabic and hebrew text
function adjustDir(inp,html) {
    if (!inp) return; inp = typeof inp === "string" ? getElement(inp) : inp; if (!inp) re-
    turn;
    var xc_arabic = '\\u0600-\\u06FF\\u0750-\\u077F\\uFB50-\\uFDFF\\uFE70-\\uFEFF';
    var xc_hebrew = '\\u0591-\\u05A0\\u05D0-\\u05EA\\u0591-\\u05A0\\u05F0-\\u05F4\\u05C0-\\u05C4';
    var xc = new RegExp('^[\u0000-\\u0020\\u2000-\\u200D]*['+xc_arabic+xc_hebrew+']');
    var cc = new RegExp('^[\u0000-\\u0020\\u2000-\\u200D]*$');
    var v = (html) ? inp.innerHTML : inp.value; if (!v) return;
    if (v.match(xc)) { // arabic character set
        inp.dir = 'rtl';
        inp.style.textAlign = 'right';
    } else if (!v.match(cc)) { // not control and punctuation character set
        inp.dir = 'ltr';
    }
}

// Ajax object
function ajax(anch) {
    this.anchname = anch;
    this.myreq = null;
    this.onerror = this.defaultError;
    this.onload = null;
    this.block = null;
    this.touttimer = null;
    this.nomsq = 0;
}

ajax.prototype.openRequest = function (url,params,httpMethod,id,onload) {
    if (!httpMethod) {
        httpMethod="POST";
    } else {
        httpMethod = httpMethod.toUpperCase();
    }
    var u = url + "?" + params;
    this.anchname = id;
    this.onload = onload;
    this.myreq = this.initXMLHttpRequest();
    if (this.myreq) {
        this.block = getElement(this.anchname);
        try {
            var loader=this;
            // Register function for readyState event
            this.myreq.onreadystatechange=function() {
                loader.onReadyState.call(loader);
            }
            this.myreq.open(httpMethod,u,true);
            var p = params;
            if (httpMethod=="POST") {
                this.myreq.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
                this.myreq.setRequestHeader("Content-length", params.length);
                this.myreq.setRequestHeader("Connection", "close");
            } else {
                this.myreq.setRequestHeader("Content-Type","text/html");
            }
            // Set timeout to 30 seconds
            this.touttimer = setTimeout(function() {
                if (this.block) {this.block.innerHTML="Request timed out!";}
                this.myreq.abort();
                OPEN_AJAX_REQUESTS--;
                if (this.onload) { req.responseText=""; this.onload.call(req.responseText); }
            }, 30000);
            // Send request
            this.myreq.send(params);
        } catch (err) {
            this.onerror.call(this);
        }
    }
}
```

```

    }
    OPEN_AJAX_REQUESTS++;
  }
}

var rscript = /<script(\s)*?\s*>/gi;

ajax.prototype.onReadyState = function() {
    var req = this.myreq;
    var ready=req.readyState;
    var data=null;
    var coord=null;
    if (this.touttimer) clearTimeout(this.touttimer);
    this.touttimer = null;
    if ( (this.block) && (!this.nomsg) ) {this.block.innerHTML="&hellip;"; }
    if (ready==READY_STATE_COMPLETE) {
        OPEN_AJAX_REQUESTS--;
        // Extract scripts, execute, and remove them from the HTML text
        if (this.block) {
            this.block.innerHTML=req.responseText.replace(rscript, "");
            var myregexp = /<script [\s\S]*?>([\s\S]*)<\s*>/;
            var match = myregexp.exec(req.responseText);
            if (match != null) {
                try {
                    eval(match[1]);
                } catch(err) {
                    // alert(err);
                }
            }
        }
        // Execute onload() function if defined
        if (this.onload) { this.onload(this); }
    }
}

ajax.prototype.initXMLHttpRequest = function() {
    var xRequest=null;
    if (window.XMLHttpRequest) {
        xRequest = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        xRequest = new ActiveXObject ("Microsoft.XMLHTTP");
    }
    return xRequest;
}

ajax.prototype.defaultError = function() {
    if (this.touttimer) clearTimeout(this.touttimer);
    this.touttimer = null;
    OPEN_AJAX_REQUESTS--;
    alert("error fetching data!"
        +"\n\nreadyState:"+this.myreq.readyState
        + "\nstatus:      "+this.myreq.status+": "+this.myreq.statusText
        + "\nheaders:    "+this.myreq.getAllResponseHeaders()
    );
}

```

8.1.2 Other scripts

The other scripts are necessary for the mmf/Editor component.

Please do not modify any of them!

8.2 Style sheets

There are a couple of style sheet files available (see 7.3 "The styles", page 30). This is the default style sheet (colour scheme *orange*):

```

.mmf_feed_panel {
    margin:0px 0px 4px 0px; padding:0;

```

```
        color:#333333; background-color:#ffffff;
        border:1px solid #990000;
    }
    .mmf_feed_header {
        margin:0px 0px 0px 0px; padding:0px 4px 2px 4px;
        color:#ffffff; background-color:#990000;
        font-size:1.2em; font-weight:bold;
        cursor:pointer;
    }
    .mmf_feed_header_link {
        color:#336699 !important;
        font-style:italic; text-decoration:none;
    }
    .mmf_feed_header_link:hover {
        color:#e0f7ff !important; background-color:#999999 !important;
        text-decoration:none;
    }
    .mmf_feed_text {
        margin:0em 0px 2px 0px; padding:2px 4px 2px 4px;
        color:#666666;
        font-size:1.0em;
    }
    .mmf_feed_text_link {
        color:#666666;
        font-style:italic; text-decoration:none;
    }
    .mmf_feed_text_link:hover {
        color:#e0f7ff !important; background-color:#999999 !important;
        text-decoration:none;
    }
    .mmf_feed_footer {
        margin:2px 0px 0px 0px; padding:0px 0px 2px 0px;
        color:#999999;
        font-size:0.8em;
    }
    .mmf_feed_link {
        margin:2px 0px 0px 0px; padding:0px 4px 0px 0px;
        color:#336699 !important;
        font-size:0.8em;
        text-decoration:none;
    }
    .mmf_feed_link:hover {
        color:#333333 !important; background-color:#e0f7ff !important;
        text-decoration:none;
    }
    .mmf_feed_footer_link {
        color:#999999;
        text-decoration:none;
    }
    .mmf_feed_footer_link:hover {
        color:#e0f7ff !important; background-color:#999999 !important;
        text-decoration:none;
    }
    .mmf_item_panel {
        margin:0px 0px 4px 0px; padding:0;
        color:#333333; background-color:#fff0cc;
        border:1px solid #cc6633;
    }
    .mmf_item_panel_last {
        margin-bottom:0;
    }
    .mmf_item_header {
        margin:0px 0px 0px 0px; padding:0px 4px 2px 4px;
        color:#ffffff; background-color:#cc6633;
        font-size:1.0em;
        font-weight:bold;
        cursor:pointer;
    }
    .mmf_item_header_link {
        color:#ffffff;
        text-decoration:none;
    }
```

```

}
.mmf_item_header_link:hover {
    color:#999999 !important; background-color:#e0f7ff !important;
    text-decoration:none;
}
.mmf_item_text {
    margin:0em 0px 2px 0px; padding:2px 4px 2px 4px;
    color:#333333;
    font-size:1.0em;
}
.mmf_item_text_link {
    color:336699 !important;
    font-style:italic; text-decoration:none;
    text-decoration:none;
}
.mmf_item_text_link:hover {
    color:#ffffff !important; background-color:#999999 !important;
    text-decoration:none;
}
.mmf_item_link {
    color:#333399 !important;
    font-style:italic; text-decoration:none;
    text-decoration:none;
}
.mmf_item_link:hover {
    color:#ffffff !important; background-color:#999999 !important;
    text-decoration:none;
}
.mmf_item_footer {
    margin:2px 0px 0px 0px; padding:0px 0px 2px 0px;
    color:#999999;
    font-size:0.8em;
}
.mmf_item_footer_link {
    color:#6699cc !important;
    text-decoration:none;
}
.mmf_item_footer_link:hover {
    color:#ffffff !important; background-color:#999999 !important;
    text-decoration:none;
}
.mmf_feed_header_link IMG,
.mmf_feed_text_link IMG,
.mmf_item_text_link IMG,
.mmf_item_link IMG {
    border: 0px;
}
.mmf_table,
mmf_tr,
mmf_td {
    margin:0; padding:0; border:0;
}
.mmf_p {
    margin:0em 0em 0.133333em 0em;
}

```

8.3 Sample web page

This is a complete web page with an embedded Feed.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<!--
    HINT: To test this code...
    - select it,
    - copy it to the clipboard,
    - in a text editor, create a new file and paste the code,
    - save it locally as (e.g.) 'test.html',
    - double click the file to show in the browser.

```

```
-->
<head>

<title>mmFeeder demo_de - en-GB</title>

<meta http-equiv="Content-Language" content="en-GB" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<!-- ##### EMBEDDING: HEADER BEGIN ##### (STEP 1) ##### -->
<link type="text/css" href="http://www.methodica.ch/mmfeed/styles/mmfeed_default.css"
rel="stylesheet" />
<script type="text/javascript"
src="http://www.methodica.ch/mmfeed/scripts/ajaxify_min.js"></script>
<script type="text/javascript">
    var nofeedhead = false; // Set to 'true' to suppress the feed header
    var refreshInterval = 5 * 60 * 1000; // Feed refresh interval (5 minutes)
    var feedTimer = 0;
    var ajaxObj = 0;

    function loadFeed() {
        if (feedTimer) {
            clearTimeout(feedTimer);
            feedTimer = 0;
        }
        ajaxObj = new ajax();
        ajaxObj.openRequest(
            'http://www.methodica.ch/mmfeed/index.php',
            'feedname=demo_de&language=en-
GB&type=html&options=inline'+((nofeedhead)?'|nohead':''),
            'GET',
            'news_panel'
        );
        feedTimer = setTimeout('loadFeed()',refreshInterval);
    }
</script>
<!-- ##### EMBEDDING: HEADER END ##### (STEP 1) ##### -->

<style type="text/css">
BODY {
    font-family:tahoma,arial,Helvetica,sans-serif;
    font-size:76%;
}

#container {
    width: 600px;
    margin: 10px auto;
    background-color: #fff;
    color: #333;
    border: 1px solid gray;
}

#header_panel {
    padding: 0.3333333em;
    background-color: #ddd;
    border-bottom: 1px solid gray;
}

#header_panel h1 {
    padding: 0;
    margin: 0;
}

#left_panel {
    float: left;
    width: 400px;
    height: 500px;
    margin: 0;
    padding: 0.3333333em;
}
```

```

#right_panel {
    height: 500px;
    margin-left: 400px;
    border-left: 1px solid gray;
    padding: 0.3333333em;
    font-size: 0.9em;
    overflow: auto;
}

#footer_panel {
    clear: both;
    margin: 0;
    padding: 0.3333333em;
    color: #333;
    background-color: #ddd;
    border-top: 1px solid gray;
}

h2 { margin: 0; padding: 0.333333em 0 0.333333em 0;}
</style>

</head>

<!-- ##### EMBEDDING: BODY BEGIN ##### (STEP 2) ##### -->
<body onload="loadFeed();">
<!-- ##### EMBEDDING: BODY END ##### (STEP 2) ##### -->

<div id="container">
    <div id="header_panel">
        <h1>Header</h1>
    </div>
    <div id="left_panel">
        <h2>Text Title</h2>
        <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla dignissim ligula
            sed turpis feugiat sit amet dignissim mi vehicula.
        </p>
        <h2>Text Title</h2>
        <p>
            Proin at mi mauris. In sollicitudin tincidunt lacus, mollis ultricies elit conse-
            quat id. Vivamus commodo placerat velit, sit amet bibendum lectus convallis eget. Etiam
            tincidunt scelerisque mauris, quis consequat mauris imperdiet vel.
        </p>
    </div>
    <div id="right_panel">
        <!-- <h2>Feed Title</h2> -->
    </div>
</div>

<!-- ##### EMBEDDING: CONTAINER BEGIN ##### (STEP 3) ##### -->
    <div id="news_panel">
        <!-- We cannot display the page correctly if JavaScript is not enabled. -->
        <noscript>
            <p style="margin: 2.0em 0em 0em 0em; font-size: 1.2em; font-weight: bold;
padding: 4px 4px 4px 4px; color: #990000; border: 2px solid #990000;">
                Note: <br>JavaScript must be enabled for this page to function prop-
            erly!<br>
                If you had JavaScript enabled you would see a message feed here.</p>
        </noscript>
    </div>
<!-- ##### EMBEDDING: CONTAINER END ##### (STEP 3) ##### -->

    </div>
    <div id="footer_panel">
        Footer
    </div>
</div>

</body>
</html>

```