MedDream DICOM Viewer

Servicing MANUAL

(version 7.1.1)
# Table of Contents

1 Introduction 1

2 Minimal server side requirements 1
   2.1 Minimal hardware requirements 1
   2.2 Minimal memory requirements 1
   2.3 Minimal software requirements 2
      2.3.1 Supported operating systems 2
      2.3.2 Supported WEB servers 2
      2.3.3 Supported PHP scripting interpreters 2

3 General considerations 2

4 Apache configuration 3
   4.1 Apache modules 3
   4.2 AllowOverride directive 4
   4.3 Reverse proxy 4

5 Deployment under Windows operating systems 4

6 Deployment under Linux operating systems 5

7 Deploying the Java-based core 7
   7.1 Configuration 7
      7.1.1 System.json 9
   7.2 Running as a service 11
      7.2.1 Windows 11
      7.2.2 Linux (System-V init) 11
      7.2.3 Linux (systemd) 11
      7.2.4 Linux (upstart) 12

8 Integration with PACS 12
   8.1 Integration modes 12
   8.2 Direct access to PACS database mode 13
      8.2.1 PacsOne 13
      8.2.2 DCM4CHEE 14
      8.2.3 ClearCanvas 15
      8.2.4 Conquest 16
      8.2.5 Aycan 17
      8.2.6 EDGE 17
   8.3 Web Access to DICOM Persistent Objects (WADO) mode 18
      8.3.1 WADO mode notes 18
      8.3.2 MedDream configuration for WADO mode 18
   8.4 DICOM mode 18
      8.4.1 DICOM mode notes 19
      8.4.2 MedDream configuration for DICOM mode 19
   8.5 File system access mode 20
      8.5.1 File system mode notes 20
      8.5.2 MedDream configuration for File system mode 20
   8.6 Plugins for Java-based core 21

9 Image Access from hospital information system (HIS) 29
   9.1 Generic HIS integration 29
   9.2 Specification of thumbnails service 31
   9.3 Specification of Live Stream 32
   9.4 Further reading 32
1. Introduction

MedDream DICOM Viewer is a HTML based package for PACS server which is designed to aid professionals in every day’s decision making process, connecting all the medical data into a unified and fast performing network. MedDream ensures a fast and reliable way to search, present and analyze the medical data (images and video files) on various devices: computers, smart phones, tablets and so forth.

MedDream covers: radiology, cardiology, oncolgy, gastroenterology and many other fields of medical application. It seamlessly integrates with various medical imaging devices, such as: ultrasound (US), magnetic resonance (MRI), positron emission tomography (PET), computed tomography (CT), endoscopy (ES), mammography (MG), digital radiography (DR), computed radiography (CR) ophthalmology, and so forth.

Core MedDream DICOM Viewer uses are:

- Replacement of hard copies, e.g. film archives, paper documents, etc.
- Remote access. MedDream provides a possibility to be mobile and work from any place in the world where the Internet is accessible. More than one person can access and view medical records at one time. Such functionality speeds up the collaboration among the professionals. So that a doctor in the hospital and a doctor that is in the different location may view the medical data and discuss about it simultaneously. The patient's medical history, various studies and images are found much faster comparing to the conventional paper-based methods.
- MedDream can be used as a standalone WEB Viewer or integrated into MedDream PACS, PacsOne PACS, dcm4chee Archive, Conquest PACS, ClearCanvas PACS systems. Moreover, MedDream can be adapted to client's PACS system and easily integrated into RIS/HIS workflow.
- MedDream has multiple functions such as search of studies, viewing, analyzing, saving, exporting, forwarding images and videos, etc.

2. Minimal server side requirements

2.1. Minimal hardware requirements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>2.33GHz or higher x64-compatible (2 CPUs)</td>
</tr>
<tr>
<td>Memory</td>
<td>8+ GB</td>
</tr>
<tr>
<td>Hard drive</td>
<td>30 GB (RAID 1, RAID 5, RAID 10)</td>
</tr>
<tr>
<td>Network Interface</td>
<td>100 Mbit/s</td>
</tr>
</tbody>
</table>

2.2. Minimal memory requirements

Minimal memory requirements for the best performance of the software:

- 8 GB of RAM if you plan to open more than 800 images (CT & MRI, PET-CT).
- 12 GB of RAM for more than 1500 images (multi-slice CT & PET-CT).
- 16 GB of RAM for more than 3000 images (cardiac or functional imaging).
2.3. Minimal software requirements

2.3.1. Supported operating systems

MedDream supports following operating systems:

- Windows Server 2008 (32 bit and 64 bit) and newer
- Windows 7 (32 and 64 bit) and newer
- Linux (32 bit and 64 bit, with glibc version >= 2.7).

2.3.2. Supported WEB servers

MedDream supports Apache HTTP server. Versions from 2.2 are supported. Under Linux operating systems distribution supplier package shall be used. Under Windows operating systems binary packages from third parties (such as XAMPP) may be used or compiled from source.

2.3.3. Supported PHP scripting interpreters

MedDream requires PHP server-side scripting language interpreter integrated into the web server as a SAPI module. PHP versions from 5.4 to 7.1 are supported. Under Linux operating systems, distribution supplier package shall be used. Under Windows operating systems, binary packages from third parties may be used or compiled from source.

Under Windows, a thread-safe (TS) 32-bit PHP build is required. The DLLs included won’t work with a 64-bit PHP, like in 64-bit builds of WampServer.

Under Linux, a non-TS PHP build is required. This is related to a more common build of Apache, the “prefork”. The “worker” build is incompatible.

The default value for memory_limit (php.ini) is often 128M or a similar small value. Large images like CR, DX and multiframe US might require a lot more. For 32-bit Apache and 2 GB of physical memory, you can start from 1536M and experiment further, as values too close to 2 GB might cause poor handling of “out of memory” situations.

MedDream will inform about missing PHP extensions that are critical to its operation. However you can try to install in advance those not listed by phpinfo() (or, in some cases, by php -m):

- required: xml, SimpleXML, json, curl and, in case of Direct access to PACS database mode, a corresponding database driver (mysqli, pgsql, etc)
- recommended: mcrypt

3. General considerations

During an upgrade it’s recommended to use the included config.sample-* .php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

After major changes in config.php (especially $pacs parameter) it might be required to restart the Java-based core; see Deploying the Java-based core.

MySQL from XAMPP 1.8 for Windows: if the web interface connects to the database very slowly (each attempt takes about 1 second), then try adding bind-address = :: to my.ini.

While still in demo mode, you must remove the existing meddream.lic after changing computer’s host name (any OS), reinstalling the OS (Windows) or changing the MAC of any network card (Linux). A correct file will be
created automatically. Otherwise a commercial license generated from your meddream.lic will not work. This also means that Windows must be activated in advance!

Under Linux, the .so file requires glibc library version 2.7 or later.

Starting from 5.5, support for DICOM overlays is on by default and MedDream will attempt to patch the image with data from the (6000,3000) tag. If some images fail to open after upgrade to 5.5+ and logs contain the string “Error” related to output of meddream_thumbnail / meddream_convert2, then try to turn off the overlays by adding meddream.overlays_enabled = 0 to php.ini.

Starting from 6.0, Java 8 is needed regardless of the PACS as parsing of some image types is implemented in a separate Java application. Later versions are increasingly replacing other functionality with Java-based implementation. See Deploying the Java-based core.

The following instruction assumes that the system already meets prerequisites and runs Apache httpd server with PHP server-side scripting interpreter.

4. Apache configuration

4.1. Apache modules

The following Apache modules must be loaded:

- mod_version
- mod_deflate
- mod_filter
- mod_proxy
- mod_proxy_http
- mod_rewrite
- mod_proxy_wstunnel

Under Windows operating systems, the following lines in the main Apache configuration file must be uncommented or added:

```plaintext
LoadModule version_module modules/mod_version.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule filter_module modules/mod_filter.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

Under Debian based systems one may use the a2enmod command:

```plaintext
a2enmod deflate
da2enmod filter
da2enmod rewrite.load
da2enmod proxy.load
da2enmod proxy_http
da2enmod proxy_wstunnel
```

a2enmod version is not needed (and will fail) if this module is compiled statically, which is often the case.

For other distributions, consult the distribution documentation.

Note: Apache service restart or reload is required after modifying httpd.conf or enabling modules by other methods.
4.2. AllowOverride directive

The `<Directory>` clause that defines MedDream installation directory (or some of its parent directories) must allow `All` or at least `FileInfo`:

```
<Directory>
    AllowOverride FileInfo
```

NOTE: the Security considerations chapter additionally suggests `Limit/AuthConfig` and `Options`. When adding them, do not drop `FileInfo`.

4.3. Reverse proxy

A reverse proxy must be set up for Java-based core. The following assumes that MedDream opens as `/meddream`:

```
ProxyPass /meddream/api/resolve !
ProxyPass /meddream/api http://127.0.0.1:8080/ nocanon
ProxyPassReverse /meddream/api http://127.0.0.1:8080/ nocanon
ProxyPass /meddream/api/study/open ws://127.0.0.1:8080/study/open
ProxyPassReverse /meddream/api/study/open ws://127.0.0.1:8080/study/open

#RewriteEngine on
#RewriteCond %{HTTP:UPGRADE} ^WebSocket$ [NC]
#RewriteCond %{HTTP:CONNECTION} Upgrade$ [NC]
#RewriteRule /meddream/api/(study/open.*) ws://127.0.0.1:8080/$1 [P]
```

The commented-out part is usually not required and should remain inactive.

If MedDream opens as `/`, then just remove the part “/meddream” from all lines. Otherwise you might need to change “/meddream” to the actual Web context path.

If you changed the Application TCP/IP port in the `application.properties` file (see Deploying the Java-based core), then “8080” above must be updated with the same value.

5. Deployment under Windows operating systems

It is assumed that Apache document root is located at `C:\XAMPP\Apache\htdocs`, and PHP is located at `C:\XAMPP\PHP` and user has administrative permissions.

1. Copy meddream folder to `C:\XAMPP\Apache\htdocs`
   (the result: `C:\XAMPP\Apache\htdocs\meddream`)
   If you are updating an older version of MedDream:
      a) Please back up the old version (so you can go back any time).
      b) Please do not replace files you changed in last version. You must carefully synchronize the new version.

2. (optional) From `C:\XAMPP\Apache\htdocs\meddream` Copy and Replace `applet.php` to `C:\XAMPP\Apache\htdocs`
   It is advised to make a backup beforehand in case you’ll need to uninstall MedDream later.

3. From `C:\XAMPP\Apache\htdocs\meddream`
   copy `php5.4_meddream.dll` to `C:\XAMPP\PHP\ext` (if you are using PHP 5.4.x)
   copy `php5.5_meddream.dll` to `C:\XAMPP\PHP\ext` (if you are using PHP 5.5.x)
copy php5.6_meddream.dll to C:\XAMPP\PHP\ext (if you are using PHP 5.6.x)
copy php7.0_meddream.dll to C:\XAMPP\PHP\ext (if you are using PHP 7.0.x)
copy php7.1_meddream.dll to C:\XAMPP\PHP\ext (if you are using PHP 7.1.x)

4. Add to php.ini file
   extension=php5.4_meddream.dll (if you are using PHP 5.4.x)
   extension=php5.5_meddream.dll (if you are using PHP 5.5.x)
   extension=php5.6_meddream.dll (if you are using PHP 5.6.x)
   extension=php7.0_meddream.dll (if you are using PHP 7.0.x)
   extension=php7.1_meddream.dll (if you are using PHP 7.1.x)

5. Specify date.timezone in php.ini:

   [Date]
   ; Defines the default timezone used by the date functions
   date.timezone = "America/Chicago"

   The list of possible values may be obtained from http://www.php.net/manual/en/timezones.php.

6. Restart Apache

7. Make sure meddream\bin\deleteTemp.bat is run at least every night by Task Scheduler. See chapter “INSTALLATION” in quick_install-Scripts.txt for a recipe. It might be needed to specify the full path to php.exe in deleteTemp.bat.

6. Deployment under Linux operating systems

Following guide assumes that Apache document root is located at /var/www/html and PHP configuration is located at /etc/php and user has root permissions.

1. Copy the “meddream” directory to /var/www/html so that the result is /var/www/html/meddream.

2. Adjust permissions: no less than

   # required for the Java Core
   find /var/www/html/services -type d -exec chmod 775 {} \;
   find /var/www/html/services -type f -exec chmod 664 {} \;

   # required for Meddream
   find /var/www/html/meddream -type d -exec chmod 775 {} \;
   find /var/www/html/meddream -type f -exec chmod 664 {} \;
   find /var/www/html/meddream -iname "*.sh" -exec chmod 775 {} \;

   # required both for PHP and Java Core
   chmod a+x /var/www/html/meddream/dcm4che/bin/*

   # required for displaying SR modality files
   chmod a+x /var/www/html/meddream/dcmtk/dsr2html

3. If you are updating an older version of MedDream:
   Please back up the old version (so you can go back any time).
   Please do not replace files you changed in last version. You must carefully synchronize the new version

4. From /var/www/html/meddream:
copy php5.4_meddream.so to /usr/lib/php/modules (if you are using PHP 5.4.x)
copy php5.5_meddream.so to /usr/lib/php/modules (if you are using PHP 5.5.x)
copy php5.6_meddream.so to /usr/lib/php/modules (if you are using PHP 5.6.x)
copy php7.0_meddream.so to /usr/lib/php/modules (if you are using PHP 7.0.x)
copy php7.1_meddream.so to /usr/lib/php/modules (if you are using PHP 7.1.x)
copy php5.4_meddream-x86_64.so to /usr/lib64/php/modules (if you are using PHP 5.4.x under a 64-bit OS)
copy php5.5_meddream-x86_64.so to /usr/lib64/php/modules (if you are using PHP 5.5.x under a 64-bit OS)
copy php5.6_meddream-x86_64.so to /usr/lib64/php/modules (if you are using PHP 5.6.x under a 64-bit OS)
copy php7.0_meddream-x86_64.so to /usr/lib64/php/modules (if you are using PHP 7.0.x under a 64-bit OS)
copy php7.1_meddream-x86_64.so to /usr/lib64/php/modules (if you are using PHP 7.1.x under a 64-bit OS)

5. Add to php.ini file

```
extension=php5.4_meddream.so (if you are using PHP 5.4.x)
extension=php5.5_meddream.so (if you are using PHP 5.5.x)
extension=php5.6_meddream.so (if you are using PHP 5.6.x)
extension=php7.0_meddream.so (if you are using PHP 7.0.x)
extension=php7.1_meddream.so (if you are using PHP 7.1.x)
extension=php5.4_meddream-x86_64.so (if you are using PHP 5.4.x under a 64-bit OS)
extension=php5.5_meddream-x86_64.so (if you are using PHP 5.5.x under a 64-bit OS)
extension=php5.6_meddream-x86_64.so (if you are using PHP 5.6.x under a 64-bit OS)
extension=php7.0_meddream-x86_64.so (if you are using PHP 7.0.x under a 64-bit OS)
extension=php7.1_meddream-x86_64.so (if you are using PHP 7.1.x under a 64-bit OS)
```

Depending from distribution PHP configuration file may be stored in different location. Please consult distribution documentation.

6. Specify date.timezone in php.ini:

```
[Date]
; Defines the default timezone used by the date functions
date.timezone = "America/Chicago"
```

The list of possible values may be obtained from http://www.php.net/manual/en/timezones.php.

7. Restart Apache

8. Make sure meddream/bin/deleteTemp.sh is run at least every night by cron. See chapter “INSTALLATION” in quick_install-Scripts.txt for a recipe. Furthermore, check if MedDream can still write to its log files after this script has been run.

9. mkisofs is required for the Export command. If your distribution does not provide mkisofs, genisoimage package may be used. In that case a symlink to genisoimage is required:

```
ln -s /usr/bin/genisoimage /usr/bin/mkisofs
```

10. To display SR files, we are using a prebuilt binary meddream/dcmtk/dsr2html from DCMTK 3.6.3. If your distribution provides another version of it that works better and still understands command line options passed by SR.php, then you’ll need to replace dsr2html and dicom.dic with symlinks to your copies.
Also please note this is a 32-bit binary, therefore some 64-bit systems might need an additional 32-bit version of glibc (for example, glibc.i386).

11. ffmpeg is required in order to display thumbnails of MPEG2 DICOM files and non-BD-compatible MPEG4 videos (Transfer Syntax UID 1.2.840.10008.1.2.4.102). If your distribution does not provide ffmpeg, avconv package may be used. In that case symlink to avconv binary is required:

```
ln -s /usr/bin/avconv /usr/bin/ffmpeg
```

12. To display videos of any kind, MedDream v5.5+ needs FFmpeg dated not older than 2014-03-08. If yours is older and can’t be updated, then you need at least to switch to the legacy behavior by changing `const USE_FFMPEG_SUBFILE = true` in flv.php to an opposite value (false).

13. To display non-BD-compatible MPEG4 videos (Transfer Syntax UID 1.2.840.10008.1.2.4.102), latest version of FFmpeg might be required.

14. Starting from 5.5, MedDream adds the option “-y” to FFmpeg command line when extracting a thumbnail from a video file. If you are forced to use an older version of FFmpeg that doesn’t understand this option, then add the following to php.ini:

```
meddream.thumb_ffmpeg_cmdline = "-i ?I -f image2 ?O"
```

### 7. Deploying the Java-based core

Starting from 6.2, the Java-based core must be running (and available via the Apache reverse proxy) for MedDream to open, even if studies are accessed via PHP.

#### 7.1. Configuration

The Java application can be found in the folder “services” of the MedDream installation archive.

1. Create a file `application.properties` in this folder (you can use the sample bundled with MedDream) and add the following configuration parameters:

   1. Location of the legacy endpoint “Routes.php”

```
com.softneta.meddream.url = MEDDREAM_URL/Routes.php
```

   Java-based core calls legacy backend for metadata etc, if so configured.

   Example: `com.softneta.meddream.url=http://127.0.0.1/Routes.php`

2. MedDream root directory

```
com.softneta.meddream.configRoot = PATH_TO_MEDDREAM_DIRECTORY
```

   Examples:

   - `com.softneta.meddream.configRoot = ../meddream`
   - `com.softneta.meddream.configRoot = /var/www/html`
   - `com.softneta.meddream.configRoot = file:///C:\\MedDreamPACS\\MedDream`

   (WARNING: If you need to use Windows paths with backslashes, then every backslash must be doubled.)

   (NOTE: the “file://” syntax might cause failure of the “Register” function; in that case you’ll need to add the line `com.softneta.license.licenseFileLocation=PATH_TO_MEDDREAM_DIRECTORY/meddream.lic`, without the “file://” part.)

3. Remote connection string and local AE Title for saving annotations and key objects
Created DICOM objects will be sent to the PACS over DICOM C-STORE. The PACS is specified by a connection string in form `AE_TITLE@HOST OR IP:PORT`. The local AET identifies MedDream. Example:

```java
com.softneta.meddream.dcmsnd.connect = DCM4CHEE@192.168.1.200:11112
com.softneta.meddream.dcmsnd.bind = MEDDREAM
```

NOTE: since 7.1, `com.softneta.meddream.dcmsnd.connect` is used after configuring "usePacsGwDirectly":false in `system.json`. Its default value, true, forces usage of plugin-local parameters .storeScpIp, .storeScpPort, .storeScpAet (see `Plugins for Java-based core`).

NOTE: in a rare event when the PACS requires client traffic to come from a certain source port, `com.softneta.meddream.dcmsnd.bind` can include hostname/IP and port in the same syntax. However since 7.1 this is possible only in the "usePacsGwDirectly":false case.

4. Under Linux, FFmpeg installation is required for video thumbnail generation.

```java
com.softneta.thumbnails.ffmpegExecutable= PATH_TO_FFMPEG
```

Under Windows, this parameter is not required; in that case it points to the bundled FFmpeg that should be sufficient.

5. Application TCP/IP port (optional)

```java
server.port = APPLICATION_PORT
```

Allows to choose a different port if the default one, 8080, is occupied. Example: `server.port=8090`.

In this case you'll also need to update the port part of `$core_addr` (config.php) to the same value.

6. Path to MedDream temporary files

```java
com.softneta.meddream.tempDir = PATH_TO_MEDDREAM_TEMP_FILES
```

Allows to specify the path to location where MedDream temp files are stored. Example: `../meddream/temp`, `/tmp/meddream`, `E:\mdtmp`.

NOTE: since 7.0 some cached files might need paths longer than 260 characters which are not allowed. The only solution is to have `com.softneta.meddream.tempDir` as short as possible.

NOTE: when specifying a path with a different Windows drive letter, always use the doubled backslashes. For example, `E:/mdtmp` and `file:/E:/mdtmp` will not work.

7. Remaining values that are reserved

```java
com.softneta.meddream.patientStudyCountEnabled=true
logging.level.com.softneta=INFO
logging.file=mdjavacore.log
```

8. PACS plugins configuration

```java
com.softneta.meddream.pacs.configurations[X].type=...
/#...
```

In most cases these can remain commented out, especially when starting MedDream for the first time. Otherwise refer to `Plugins for Java-based core` for details.
2. Start the Java application manually with a console command “java -jar”, for example, java -jar MedDream-7.1.jar.

**NOTE:** for a more comprehensive support of compressed DICOM transfer syntaxes, the native libraries bundled with MedDream must be added when starting the application on Windows, or on Linux:

1. open libs.zip and choose the folder suitable for your operating system. For example, win-i686;
2. extract file(s) from that folder to a new folder of your choice. For example, C:\MedDreamPACS-Premium\services\lib;
3. specify the path to this folder in the command that starts Java: -Djava.library.path=PATH_TO_FOLDER. Also the option -jar changes to -cp and another option org.springframework.boot.loader.PropertiesLauncher is added. For example, java -cp MedDream-7.1.jar -Djava.library.path=C:\MedDreamPACS-Premium\services\lib org.springframework.boot.loader.PropertiesLauncher.

You should first make sure everything works when the application is started manually. Then follow the next chapter for a more standalone setup.

**IMPORTANT:** If the Query/Retrieve functionality is used:

- **PHP:** The tempDir must use the default value.
- **QR plugin:** tempDir and cacheDir must be the same.

### 7.1.1. System.json

The file system.json (meddream/settings/system.json by default) is automatically created if missing when the Java-based core starts. Several key features require this file to be updated; in most cases the core must be restarted afterwards.

An example of the entire file is shown below.

```
{
    "studyMinDate": "19000101",
    "studyMaxDate": "20000101",
    "usePacsGwDirectly": true,
    "disableMultiFrameVideoAutoLoad": false,
    "features": {
        "patientHistory": true,
        "keyObjects": true,
        "presentationState": true,
        "searchSettings": true,
        "viewerSettings": true,
        "reportSettings": true,
        "pluginsSettings": false,
        "hangingProtocolsSettings": true
    },
    "threeDimensionServiceIntegration": {
        "port": "8089",
        "protocol": "https:",
        "serverIP": "127.0.0.1",
        "pathPrefix": "D:\DICOM",
        "pathPrefixReplaceTo": "Z:\DICOM"
    },
    "blackListedSopClasses": ["1.2.840.10008.5.1.4.1.1.66", "1.2.840.10008.5.1.4.1.1.4.2"],
    "forwardPacs": {
        "storeScpIp": "127.0.0.1",
        "storeScpPort": 104,
    }
}
```
• studyMinDate: “19900101”

  (PHP only) Default “from” date in YYYYMMDD format when empty in the user interface. Might be important with $pacs='DICOM' and some destination PACSes; might also minimize surprises in any kind of integration with a too slow search.

• studyMaxDate: “20000101”

  (PHP only) Default “to” date in YYYYMMDD format when empty in the user interface.

• usePacsGwDirectly: true

  The frontend will call Java backend instead of the legacy PHP backend.

• disableMultiFrameVideoAutoLoad: false

  (reserved) Disables multiframe and video autoload. Used for automated testing purposes, do not change this setting.

• patientHistory: true

  The viewer window will load patient history.

• keyObjects: true

  The viewer window will load key objects.

• presentationState: true

  The viewer window will load presentation state objects.

• searchSettings: true

  Enable search settings page in the settings window.

• viewerSettings: true

  Enable viewer settings page in the settings window.

• reportSettings: true

  Enable report settings page in the settings window.

• pluginsSettings: false

  true enables plugins store settings page in the settings window. (If enabled, only plugins installed from api storage will be loaded, otherwise all plugins will be loaded.)

• hangingProtocolsSettings: true

  Enable hanging protocol settings page in the settings window.

• blackListedSopClasses: ‘[“1.2.840.10008.5.1.4.1.1.66”]’

  Thumbnails of images with these SOP Classes will not be shown in the viewer.

• storeScpIp: “127.0.0.1”

  (planned) Destination IP address for the Forward function.

• storeScpPort: 4242

  (planned) Destination TCP port for the Forward function.

• storeScpAet: “PACS”

  (planned) Destination AE Title for the Forward function.
• description: “An example destination for Forward”

(planned) Human-readable description of a destination device for the Forward function.

7.2. Running as a service

7.2.1. Windows

1. In the directory of the Java application, copy MedDream.NET2.exe or MedDream.NET4.exe (depends on installed .NET version) to MedDream.exe;
2. Open the Command Prompt;
3. Run the command MedDream.exe install;
4. Start the service from the Windows service manager.

NOTE: MedDream.NET2.exe is used in case the .NET Framework 2.0 is installed meanwhile the MedDream.NET4.exe is used for .NET Framework 4.0.

NOTE: if you need a native library from “libs.zip” mentioned earlier, then update <arguments>...</arguments> in MedDream.xml accordingly.

7.2.2. Linux (System-V init)

If your architecture is not 64-bit but 32-bit, then rename {INSTALL_DIRECTORY}/meddream-jar-wrapper.i386 to meddream-jar-wrapper.

If your system is RedHat-like (Fedora, RHEL, etc), then copy {INSTALL_DIRECTORY}/meddream.redhat to {INSTALL_DIRECTORY}/meddream. If it’s Debian-like (Ubuntu, Debian itself, etc), then copy {INSTALL_DIRECTORY}/meddream.debian to {INSTALL_DIRECTORY}/meddream.

Update this line in {INSTALL_DIRECTORY}/meddream with real installation directory:

```
BASEDIR={INSTALL_DIRECTORY}
```

Execute the following commands:

```
sudo ln -s {INSTALL_DIRECTORY}/meddream /etc/init.d/meddream
sudo chmod +x {INSTALL_DIRECTORY}/meddream {INSTALL_DIRECTORY}/meddream-jar-wrapper
sudo service meddream start
sudo chkconfig meddream on
```

As Debian-like systems don’t have chkconfig, the last command can be replaced with sudo update-rc.d meddream defaults.

By changing the line USER=root in {INSTALL_DIRECTORY}/meddream, you can force a less privileged user that can still access DICOM files etc. However this will likely require updating the line PIDF=/var/run/$PROG.pid, too, as /var/run is usually a privileged location (a chown’ed subdirectory will suffice in that case).

The file {INSTALL_DIRECTORY}/meddream also contains the JRE command line, for example:

```
MD_JAR_COMMAND="java -cp MedDream-7.1.jar -Djava.library.path=./lib org.springframework.boot.loader.PropertiesLauncher"
```

7.2.3. Linux (systemd)

Create the file /etc/systemd/system/meddream.service with the following text:
[Unit]
Description=MedDream Viewer
After=syslog.target

[Service]
WorkingDirectory=/opt/meddream/services/
User=www-data
ExecStart=/usr/bin/java -Xmx1G -cp /opt/meddream/services/MedDream-7.1.0.jar -Djava.library.path=/opt/meddream/services/lib org.springframework.boot.loader.PropertiesLauncher
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target

Execute the following command for service autostart:

```
sudo systemctl enable meddream.service
```

7.2.4. Linux (upstart)

Create a file `/home/{user name}/.config/upstart/meddream.conf` with the following text:

```
description "MedDream Viewer"
respawn
exec java -Xmx1G -cp {INSTALL_DIRECTORY}/MedDream-7.1.jar -Djava.library.path={INSTALL_DIRECTORY}/lib org.springframework.boot.loader.PropertiesLauncher
```

8. Integration with PACS

8.1. Integration modes

MedDream can access studies from PACS using:

<table>
<thead>
<tr>
<th>Method</th>
<th>PACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct access to PACS database</td>
<td>PacsOne (<a href="http://pacsonone.net/">http://pacsonone.net/</a>)</td>
</tr>
<tr>
<td></td>
<td>DOM4CHEE (<a href="https://www.dcm4che.org/">https://www.dcm4che.org/</a>)</td>
</tr>
<tr>
<td></td>
<td>Conquest (<a href="https://ingenium.home.xs4all.nl/dicom.html">https://ingenium.home.xs4all.nl/dicom.html</a>)</td>
</tr>
<tr>
<td></td>
<td>ClearCanvas ImageServer (<a href="https://www.clearcanvas.ca/">https://www.clearcanvas.ca/</a>)</td>
</tr>
<tr>
<td></td>
<td>Aycan (<a href="https://aycan.com/clinical-solutions/aycan-store">https://aycan.com/clinical-solutions/aycan-store</a>)</td>
</tr>
<tr>
<td></td>
<td>EDGE (<a href="http://www.ashvatech.com/the-edge-pacs.php">http://www.ashvatech.com/the-edge-pacs.php</a>)</td>
</tr>
<tr>
<td>Web Access to DICOM Persistent Objects (WADO)</td>
<td>Any PACS that supports the WADO-URI service</td>
</tr>
<tr>
<td>DICOM 3.0 native interface</td>
<td>Any PACS</td>
</tr>
<tr>
<td>File system access</td>
<td>Not applicable (a PACS is not necessary)</td>
</tr>
</tbody>
</table>

For best performance, it is recommended to use direct access to PACS database whenever available.

**WARNING:** In the Direct access to PACS database mode, MedDream must either:

- be installed on the same host as PACS, or
- have access to studies’ files.
When MedDream is installed on a different host, the following problem might arise. Some databases contain paths to DICOM files stored on the PACS host (not on some common network storage, etc) and it is not always possible to create identical mount points or network drive letters on the MedDream host. In this case adding the parameter $foreign_path_mapping (config.php) will help. Example:

```php
$foreign_path_mapping = "*/volume1/DICOMNL\|O:\n/volume2/DICOMNL\|P:\";
```

Here /volume1/DICOMNL\ is replaced with O:\ under which the network resource shared by a Linux system is mounted on a Windows system; in the same fashion /volume2/DICOMNL\ is replaced with P:\. For best results, please take a look at php-YYYYMMDD.log for original paths (see Troubleshooting).

In Java-based core this needs to be configured separately at plugin level. Most direct access plugins have an equivalent setting mappedStorageLocation with a slightly different syntax.

### 8.2. Direct access to PACS database mode

#### 8.2.1. PacsOne

**8.2.1.1. PacsOne notes**

**WARNING:** PacsOne and Apache must run as the same user because newer versions of PacsOne create subdirectories with permissions too strict for different users. Please ensure that on Linux operating systems PacsOne and Apache services use the same user or belong to the same group.

**WARNING:** PacsOne must store the received DICOM files in the “DicomPart10” format. The other format, “Native”, is not fully supported by HTML5 MedDream (and you might have problems with other DICOM-related software). This setting is chosen during installation and later can be verified as follows:

- **Windows**: in the registry, REG_SZ StorageFormat under HKEY_LOCAL_MACHINE\SOFTWARE\RainbowFish Software\PacsOne\${AeTitle}
- **Linux**: the setting StorageFormat in file(s) *.ini near PacsOne.exe

Under Linux, PacsOne's custom libjpeg conflicts with standard libjpeg needed by the GD2 extension on which MedDream might still depend. You will need to disable the antispam image in order to log in to PacsOne web interface (to do that, place an empty file “no.antispam.code” near PacsOne.exe). You also won’t be able to view DICOM images in PacsOne; on the other hand, MedDream adds a “Show” button to the web interface and thus can be used instead of the built-in viewer.

PacsOne doesn’t have an index on series.modality. If searches with a particular modality are unacceptably slow compared to the “All” choice, then you’ll need to use the following MySQL command:

```
ALTER TABLE series ADD INDEX (modality);
```

For the Send to DICOM Library Anonymizer, Java 7 and up is required.

Since 3.06.1103.3003, a DICOMDIR viewer in the subdirectory “DICOMDIR” is automatically included on export. However, an alternative way is to use PacsOne's "Include external viewer" functionality. If you use the latter, and still keep the viewer in the subdirectory “DICOMDIR”, MedDream Export function might fail due to same files included twice. The solution is to rename the subdirectory, or move it somewhere outside the MedDream directory tree.

Until 6.0, the HIS integration by Patient ID was using a certain kind of fuzzy matching that includes coerced values of Patient ID. For example, /?patient=12345 will also list studies with Patient ID “12345[some_original_value]”. In 6.0 this is turned off by default. If you still need the legacy behavior, then change STRICTLY_MATCH PATIENT FROM ACTION in meddream/pacs/PacsImplPacsone/Search.php as follows:

```php
const STRICTLY_MATCH PATIENT FROM ACTION = false;
```

Since MySQL 5.7, the ONLY_FULL_GROUP_BY mode is on by default. However MedDream’s Search function uses an incompatible query by default. We advise to disable ONLY_FULL_GROUP_BY. If that’s not possible,
then an updated and a much slower query can be switched on by changing NO_MODALITY_AGGREGATE in meddream/pacs/PacsImplPacsone/Search.php as follows:

```php
const NO_MODALITY_AGGREGATE = false;
```

During an upgrade it’s recommended to use the included config.sample-pacsone.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

### 8.2.1.2. MedDream configuration for PacsOne

1. Perform steps described in paragraphs either *Deployment under Windows operating systems* or *Deployment under Linux operating systems*. It is recomended to use PACSONE_INSTALL_DIR/php as Apache’s DocumentRoot.

2. From PACSONE_INSTALL_DIR/php/meddream Copy and Replace applet.php to PACSONE_INSTALL_DIR/php.


4. Make sure the Java-based core is running (see *Deploying the Java-based core*) and the reverse proxy is configured in Apache (see *Apache configuration*).

5. Navigate to http://127.0.0.1/pacsone/meddream/home.php (use PacsOne users to log in) or use the “Show” button in the PacsOne web system.

   If you get an HTTP 500 error, please check *Apache configuration* once more as likely some Apache module is missing.

### 8.2.2. DCM4CHEE

#### 8.2.2.1. DCM4CHEE notes

**WARNING:** compatibility is tested only with DCM4CHEE 2.4.17, 4.3.0-Alpha4 and 5.6.0. Some (but not all) newer 5.x versions are tested occasionally.

**WARNING:** DCM4CHEE v4 and v5 (dcm4chee-arc) have been tested with MySQL and Oracle only.

**WARNING:** DCM4CHEE v2 has been tested with MySQL, MS SQL Server and PostgreSQL only.

**WARNING:** the integrated Reports functionality is so far implemented only for MySQL and Oracle. There are no schema files for MSSQL or PostgreSQL and existing queries are incompatible with these DBMSes.

For the Send to DICOM Library Anonymizer, Java 7 and up is required.

DCM4CHEE v4 requires that the path prefix (filesystem.fs_uri, default value specified in LDAP as dcmInitFileSystemUri) contains a URI scheme. MedDream supports the following syntax:

<table>
<thead>
<tr>
<th>URI</th>
<th>Path</th>
<th>Applicable to operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>file:///dir/subdir1</td>
<td>/dir/subdir1</td>
<td>all</td>
</tr>
<tr>
<td>file://dir/subdir2</td>
<td>/dir/subdir2</td>
<td>all</td>
</tr>
<tr>
<td>file:///&lt;drive&gt;:/subdir1</td>
<td>&lt;drive&gt;:/subdir1</td>
<td>all</td>
</tr>
<tr>
<td>file://&lt;drive&gt;:/subdir2</td>
<td>&lt;drive&gt;:/subdir2</td>
<td>all</td>
</tr>
<tr>
<td>file://host/share/dir</td>
<td>//host/share/dir</td>
<td>Windows</td>
</tr>
<tr>
<td>file://host/share/dir</td>
<td>//host/share/dir</td>
<td>Windows</td>
</tr>
<tr>
<td>file:///host/share/dir</td>
<td>//host/share/dir</td>
<td>Windows</td>
</tr>
</tbody>
</table>

MedDream is able to automatically include a DICOMDIR viewer in each disc of exported studies. To use that, simply place the viewer files in a subdirectory named “DICOMDIR”. You must also ensure that a file named “autorun.inf” exists in that directory, otherwise contents won’t be included; an empty autorun.inf is suitable if the viewer somehow doesn’t contain it.

MedDream WEB DICOM Viewer 7.1.1 Servicing Manual (document version 1.0)
During an upgrade it’s recommended to use the included config.sample-dcm4chee.php or config.sample-dcm4chee-arc.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

### 8.2.2.2. MedDream configuration for DCM4CHEE

1. Perform steps described in paragraphs either Deployment under Windows operating systems or Deployment under Linux operating systems.
3. Add “studynotes” table to DCM4CHEE database from the schema file reports-dcm4chee-mysql.sql if you’re using MySQL. For dcm4chee-arc/dcm4chee-arc-lite and Oracle, use reports-dcm4chee-arc-oracle.sql instead.
5. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).
6. Navigate to http://127.0.0.1/meddream/home.php and use database accounts to log in. With DCM4CHEE 2.x, its internal accounts like “admin” or “user” are supported, too.

   If you get an HTTP 500 error, please check Apache configuration once more as likely some Apache module is missing.

### 8.2.3. ClearCanvas

#### 8.2.3.1. ClearCanvas notes

**WARNING:** ClearCanvas support was last tested with MedDream 5.4. Pre-release tests with $pacs = "ClearCanvas" are rarely performed. Please contact info@softneta.com if you would like to use this integration mode.

**WARNING:** only the older ClearCanvas v2 is supported. We didn’t test MedDream with ClearCanvas 3+.

Direct integration into IIS is not possible any more, as MedDream still does not support CGI/FastCGI, ISAPI support was removed in PHP 5.3 and we dropped support for 5.2. php5.2_meddream.dll from an older version will not work. Your only option is to host MedDream in Apache (running on a different port) and use IIS as a reverse proxy. Please contact info@softneta.com for instructions.

The traditional php_mssql.dll is officially deprecated and might not work with newer versions of PHP and SQL Server. Its replacement, the “Microsoft SQL Server Driver for PHP” ($dbms = 'SQLSRV' in config.php), can be downloaded from Microsoft free of charge. It contains many flavors of the driver; the ones supported by MedDream are non-PDO and TS. The v2.0 driver also requires the “Microsoft SQL Server 2008 R2 Native Client”. Drivers older than v1.1 might be incompatible.

MedDream is able to automatically include a DICOMDIR viewer in each disc of exported studies. To use that, simply place the viewer files in a subdirectory named “DICOMDIR”. You must also ensure that a file named “autorun.inf” exists in that directory, otherwise contents won’t be included; an empty autorun.inf is suitable if the viewer somehow doesn’t contain it.

During an upgrade it’s recommended to use the included config.sample-clearcanvas.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

---

MedDream WEB DICOM Viewer 7.1.1 Servicing Manual (document version 1.0)
8.2.3.2. MedDream configuration for ClearCanvas

1. Perform steps described in the paragraph Deployment under Windows operating systems.

   You will likely need to specify a port for database connection, like localhost,1433.

3. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).

4. Navigate to http://server IP address:Apache port/meddream/home.php (do not forget the actual port that Apache runs on, and use database accounts to log in).
   If you get an HTTP 500 error, please check Apache configuration once more as likely some Apache module is missing.

8.2.4. Conquest

8.2.4.1. Conquest notes

MedDream supports Conquest only with MySQL and SQLite v3 databases at the moment.

MedDream does not support the “V2 (allows NKI compression)” image format. All images that were received by Conquest with this setting on, including those of the example patient “HEAD EXP2”, will be unusable.

For the Send to DICOM Library Anonymizer, Java 7 and up is required.

MedDream is able to automatically include a DICOMDIR viewer in each disc of exported studies. To use that, simply place the viewer files in a subdirectory named “DICOMDIR”. You must also ensure that a file named “autorun.inf” exists in that directory, otherwise contents won’t be included; an empty autorun.inf is suitable if the viewer somehow doesn’t contain it.

During an upgrade it’s recommended to use the included config.sample-conquest.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

8.2.4.2. MedDream configuration for Conquest

1. Perform steps described in paragraphs either Deployment under Windows operating systems or Deployment under Linux operating systems.


3. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).

   If you get an HTTP 500 error, please check Apache configuration once more as likely some Apache module is missing.

5. For more supported image types, see Deploying the Java-based core.
8.2.5. Aycan

8.2.5.1. Aycan notes

MedDream is able to automatically include a DICOMDIR viewer in each disc of exported studies. To use that, simply place the viewer files in a subdirectory named “DICOMDIR”. You must also ensure that a file named “autorun.inf” exists in that directory, otherwise contents won’t be included; an empty autorun.inf is suitable if the viewer somehow doesn’t contain it.

During an upgrade it’s recommended to use the included config.sample-aycan.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

Installations of aycan store use Nginx as a web server. MedDream, however, does not support the FastCGI mechanism by which PHP integrates into Nginx. The following is needed (please contact info@softneta.com for detailed instructions):

- install Apache,
- run it on different ports (for example, 82 and 8443),
- add its user to the usergroup of the PACS process for access to DICOM files (for example, adduser www-data users),
- (optional) set up a reverse proxy in Nginx for more convenient URLs.

8.2.5.2. MedDream configuration for Aycan

1. Perform steps described in the paragraph Deployment under Linux operating systems.
3. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).
4. Navigate to http://server IP address:Apache port/meddream/home.php (do not forget the actual port that Apache runs on, and use database accounts to log in).

From a security standpoint MedDream should use dedicated less-privileged database accounts. Minimum privileges are: SELECT on tables “stustore” and “serstore”, SELECT and UPDATE on the table “session”.

8.2.6. EDGE

8.2.6.1. EDGE notes

During an upgrade it’s recommended to use the included config.sample-edge.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

8.2.6.2. MedDream configuration for EDGE

1. Perform steps described in the paragraph Deployment under Windows operating systems. Do not forget to install Microsoft SQL Server Driver for PHP (compatible with your version of PHP); the installation instructions will also tell which version of Microsoft SQL Native Client is needed.
3. Make sure the Java-based core is running (see *Deploying the Java-based core*) and the reverse proxy is configured in Apache (see *Apache configuration*).

4. The Java-based core must be configured with the EDGE plugin, as the PHP part of MedDream is used only for authentication and doesn’t implement access to studies. Restart Java after the changes.

5. Navigate to http://server IP address:Apache port/meddream/home.php (do not forget the actual port that Apache runs on, and use PACS internal accounts to log in).

### 8.3. Web Access to DICOM Persistent Objects (WADO) mode

MedDream uses the dcmqr utility from dcm4che2 Toolkit to make all queries to the target PACS. This way any PACS that talks DICOM protocol is supported. DICOM images, however, are downloaded through WADO-URI interface so the target PACS must support it. Additionally, allow_url_fopen (php.ini) must remain enabled.

#### 8.3.1. WADO mode notes

For the Send to DICOM Library Anonymizer, at least Java 7 is required.

MedDream is able to automatically include a DICOMDIR viewer in each disc of exported studies. To use that, simply place the viewer files in a subdirectory named “DICOMDIR”. You must also ensure that a file named “autorun.inf” exists in that directory, otherwise contents won’t be included; an empty autorun.inf is suitable if the viewer somehow doesn’t contain it.

During an upgrade it’s recommended to use the included config.sample-wado.php as a template and then update the resulting config.php according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

#### 8.3.2. MedDream configuration for WADO mode

1. Perform steps described in paragraphs either *Deployment under Windows operating systems* or *Deployment under Linux operating systems*.


3. Make sure the Java-based core is running (see *Deploying the Java-based core*) and the reverse proxy is configured in Apache (see *Apache configuration*).

4. Navigate to http://127.0.0.1/meddream/home.php and simply press the “Login” button. There is no password. Any non-empty user name is suitable. The user name “root” additionally enables the settings button.

   If you get an HTTP 500 error, please check *Apache configuration* once more as likely some Apache module is missing.

5. Usually the target PACS must be additionally configured to accept queries from the AET named “MEDDREAM”. You can assign any port, even the default 104, as dcmqr normally does not bind to a particular port and most PACSes do not require one when accepting a connection.

### 8.4. DICOM mode

MedDream uses the dcmqr utility from dcm4che2 Toolkit to make all queries to the target PACS. DICOM images are downloaded in a similar fashion through the dcmrcv utility. This way any PACS that talks DICOM protocol is supported.
8.4.1. DICOM mode notes

dcmqr identifies itself as “MEDDREAM” and is not required to bind to a particular port (though it’s able to do that). dcmrcv must bind to a port (we recommend 11116) and for simplicity it uses the same AET, “MEDDREAM”. In general,

- the target PACS must be configured to accept this AET on port 11116,
- $db_host should contain only the AET (without IP and port),
- $dcm4che_recv_aet must contain a full connection string with AET and port.

The dcmrcv utility will automatically start (if needed) in the background every time when someone logs in. Afterwards it is managed by a particular Apache child process that started it. When this process finishes, dcmrcv will stop as well, therefore concurrent sessions might suddenly encounter the “C-MOVE operation failed” error. This normally occurs only when stopping the entire web server. You should leave MaxConnectionsPerChild (httpd.conf) with a default value (zero), or set it to a large enough value, to maximize the lifetime of dcmrcv.

Since 5.x, the RetrieveEntireStudy mode is on by default. When opening a study, if at least one image is missing from the cache, the entire study is downloaded in the background – this is a lot faster. In case the old behavior (download a single image at once) is preferred, you can switch it back on:

1. In php.ini, change or add: meddream.retrieve_entire_study = 0 (the default value is 1)
2. Restart Apache.

Only a limited support for Big Endian files is present. It causes some PACSes to consider the A-ASSOCIATE-AC response from dcmrcv invalid, and no files can be received at all. In that case your only option is to turn Big Endian support off, by adding the following to php.ini:

```
$meddream.dcmrcv_cmdline="?1 -dest cached?2"
```

8.4.2. MedDream configuration for DICOM mode

1. Perform steps described in paragraphs either Deployment under Windows operating systems or Deployment under Linux operating systems.

- $db_host and $dcm4che_recv_aet may contain the same AET. The @IP:PORT part must be present in $dcm4che_recv_aet only. It is better to avoid it in $db_host, however if you need it, then give a different value.
3. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).
4. Navigate to http://127.0.0.1/meddream/home.php and simply press the “Login” button. There is no password. Any non-empty user name is suitable. The user name “root” additionally enables the settings button.

   If you get an HTTP 500 error, please check Apache configuration once more as likely some Apache module is missing.
5. Usually the target PACS must be additionally configured to accept queries from, and send images to, the AET named “MEDDREAM” on port 11116. If that port is already used on your system, choose a different one and specify it in $dcm4che_recv_aet.
6. Some PACSes require that a Study Date search key is sent on every C-FIND request. You can configure a sufficiently wide range of dates in meddream/settings/system.json, parameters are called studyMinDate and studyMaxDate (date separators are not needed). For example,
In the Java "Query/Retrieve" plugin, the equivalent settings are named \textit{defaultStartDate} and \textit{defaultEndDate}.

8.5. File system access mode

This MedDream mode allows to view single DICOM files, or directories containing files of a single study. Limitations are as follows:

- The path to a single file, or to a directory, will be relative. Absolute paths are not allowed due to security concerns. It is also not recommended to specify a drive root as a base directory. The simplest form of HIS integration results in paths visible for everyone, hence integration via POST requests is advised.

- If a directory is specified, then DICOM files are collected directly below it (subdirectories are not supported). Contents of the directory are sorted by name in ascending order and then parsed. All files must have the same Study UID; the value in the first file is treated as a reference and remaining files with different UIDs are ignored. Consequently, if the directory contains multiple studies, the outcome depends solely on file names and their collation according to the system-default locale.

The Java "Filesystem" plugin supports subdirectory scanning. On the other hand, currently it doesn’t support copies of the same DICOM file (same Study/Series/SOP Instance UIDs) in different directories.

- There is no search functionality. A HIS (or equivalent application) must track studies/images together with corresponding paths, and offer hyperlinks to MedDream. The interactive login, however, can still be used for the Settings dialog, provided that you log in as “root” (no password required). Alternatively, you can enable the dialog for all users, by changing \texttt{SHOW\_USER} in \texttt{external.php} to “root”.

- The “Save Images...” function does not support the “Active Series” choice. However the user interface won’t warn about this, you’ll simply get images.zip which is a text file and contains an error message.

8.5.1. File system mode notes

For the Send to DICOM Library Anonymizer, at least Java 7 is required.

During an upgrade it's recommended to use the included \texttt{config.sample-filesystem.php} as a template and then update the resulting \texttt{config.php} according to the old one. This will prevent misunderstandings like variable names which were suddenly changed.

8.5.2. MedDream configuration for File system mode

1. Perform steps described in paragraphs either \textit{Deployment under Windows operating systems} or \textit{Deployment under Linux operating systems}.

2. Edit \texttt{APACHE\_HTDOCS\_DIR/meddream/config.php} file. Use \texttt{config.sample-filesystem.php} as a template. The file contains basic instructions.

   You must specify $archive\_dir\_prefix – a base directory for all allowed paths.
3. Make sure the Java-based core is running (see Deploying the Java-based core) and the reverse proxy is configured in Apache (see Apache configuration).

4. Test the functionality by browsing to

   http://127.0.0.1/meddream/?file=PATH_TO_YOUR_FILE

   For example, the directory C:\PACS\dc4chee\server\default\archive contains DCM4CHEE 2.x archive tree, with deeper levels named like year\month\day\hour\. ... This directory is specified by $archive_dir_prefix. Then PATH_TO_YOUR_FILE could be 2012\2\4\0\43D7AA94\2569DF62\9242C40A or, in JavaScript code, 2012
   \2\4\0\43D7AA94\2569DF62\9242C40A.

   If you get an HTTP 500 error, please check Apache configuration once more as likely some Apache module is missing.

8.6. Plugins for Java-based core

Support for PACSes is being added to Java, in form of PACS plugins. Since 7.1 it is enabled by default; if a particular installation still has problems, MedDream can be switched back to PHP with "usePacsGwDirectly":false in the file meddream/settings/system.json (created automatically during the first startup).

Multiple PACSes can be defined, then used in the Search window to open studies. Examples of configuration options are listed below.

NOTE: Some functions (namely, Export/Burn, Forward and Reporting, also some aspects of HIS integration) are still PHP-based at the moment. Consequently, their support depends solely on $pacs in config.php as before. Multiple PACSes will introduce additional issues: for example, the study opened from a different PACS might be missing on the PACS that is configured at PHP side.

NOTE: in 7.1, to save Key Objects and Presentation States (annotations), the following parameters must be added to the configuration of every plugin:

- Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].storeScpIp=127.0.0.1
  IP of the Storage Server.

- Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].storeScpPort=104
  Storage Server Port.

- Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].storeScpAet=PACSONE
  Storage Server AE Title.

The local AE Title is configured by com.softneta.meddream.dcmsnd.bind as before.

In some cases it might still be needed to use the legacy PHP implementation together with "usePacsGwDirectly":true (for example, no PACS plugins are currently available for ClearCanvas and Conquest). Then the abovementioned parameters can be added as follows:

```java
com.softneta.meddream.pacs.configurations[0].type=php
com.softneta.meddream.pacs.configurations[0].id=php
com.softneta.meddream.pacs.configurations[0].storeScpIp=127.0.0.1
com.softneta.meddream.pacs.configurations[0].storeScpPort=104
com.softneta.meddream.pacs.configurations[0].storeScpAet=PACSONE
```

1. PacsOne plugin

   - Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].type=PacsOnePlugin
Use this specific value of `type` when connecting to PacsOne.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].id=PacsOne`
  
  Identifies the plugin/PACS pair in the search window drop-down menu. You can connect to multiple PacsOne instances by using different values of `id` and remaining options.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].searchApiEnabled=true`
  
  Specifies if this plugin/PACS pair returns a study list for the Search window and provides study metadata.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].imageApiEnabled=true`
  
  Specifies if this plugin/PACS pair retrieves image data.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].url=jdbc:mysql://127.0.0.1:3306/PACS_DB_NAME`
  
  JDBC connection string. Only MySQL is supported.

  **NOTE:** SSL requirements can be disabled adding the following arguments:

  - `verifyServerCertificate=false` Disables Server certificate verification.
  - `useSSL=false` Disables SSL usage.
  - `requireSSL=false` Disables the SSL requirement.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].username=UserName`
  
  Username for connecting to the database.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].password=UserPassword`
  
  Password for connecting to the database.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].mappedStorageLocation=D:/DICOM=Y:\DICOM1|E:/DICOM=Y:\DICOM2`
  
  (optional) Defines replacement paths for MedDream installed on a different host than the PACS and accessing images through mapped network drives.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].useModalityAggregation=true`
  
  Defines if modalities will be aggregated in database query results. Aggregation ensures that all modalities are visible in the Modality column but this could cause performance degradation. If set to false, then SQL mode must not contain ‘ONLY FULL GROUP BY’. More information about SQL mode can be found at: https://dev.mysql.com/doc/refman/8.0/en/sql-mode.html.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].strictSearchIsEnabled=false`
  
  `true` enables exact matching by Patient ID in the Search window.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].defaultStartDate=20181107`
  
  (optional) Default “from” date in `YYYYMMDD` format when empty in the user interface.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].defaultEndDate=20181108`
  
  (optional) Default “to” date in `YYYYMMDD` format when empty in the user interface.
• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  encoding=ISO-8859-1
  Encoding database string values. encoding=ISO-8859-1 then value will be encode UTF-8 ->
  ISO-8859-1

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  encodingTo=ISO-8859-8
  Decoding database string values. If encoding=ISO-8859-1 and encodingTo=ISO-8859-8, then result will be decoded ISO-8859-1 -> ISO-8859-8; however if encoding option is empty then result will be decoded from default JVM system encoding to ISO-8859-8.

2. DCM4CHEE 2.x plugin

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  type=Dcm4chee2Plugin
  Use this specific value of type when connecting to DCM4CHEE 2.x.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  id=Dcm4chee2
  Identifies the plugin/PACS pair in the search window drop-down menu. You can connect to multiple DCM4CHEE instances by using different values of id and remaining options.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  searchApiEnabled=true
  Specifies if this plugin/PACS pair returns a study list for the Search window and provides study metadata.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  imageApiEnabled=true
  Specifies if this plugin/PACS pair retrieves image data.

• Examples: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  url=jdbc:mysql://127.0.0.1:3306/PACS_DB_NAME, url=jdbc:postgresql://127.0.0.1:5432/PACS_DB_NAME
  JDBC connection string. Supported DBMSes are MySQL and PostgreSQL.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  username=UserName
  Username used to connect to the database.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  password=UserPassword
  Password used to connect to the database.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  charset=utf8
  (optional) Defines the character set used.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  enableStatistics=false
  (optional) true enables query statistics such as query time and so on. You might also want to set logging.level.org.hibernate.stat=DEBUG in application.properties to get more details.

3. DCM4CHEE 5 (dcm4chee-arc-light) plugin

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  type=Dcm4chee5Plugin
  Use this specific value of type when connecting to DCM4CHEE 5.
• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
id=Dcm4chee5

Identifies the plugin/PACS pair in the search window drop-down menu. You can connect to multiple DCM4CHEE instances by using different values of id and remaining options.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
searchApiEnabled=true

Specifies if this plugin/PACS pair returns a study list for the Search window and provides study metadata.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
imageApiEnabled=true

Specifies if this plugin/PACS pair retrieves image data.

• Examples: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
0.0.1:5432/PACS_DB_NAME

JDBC connection string. Supported DBMSes are MySQL and PostgreSQL.

In case of connection problems it might be necessary to add some more parameters to this string: useJDBCCompliantTimezoneShift=true for a time zone shift, serverTimezone=UTC for a specific timezone, useSSL=false to disable SSL, etc. (The first parameter-value pair is preceded by “?”; the remaining pairs – by “&”.)

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
username=UserName

Username used to connect to the database.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
password=UserPassword

Password used to connect to the database.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
dialect=MySQL

Set database dialect. Supported values are MySQL and PostgreSQL. (Somewhat duplicates the url parameter, this is by design.)

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
encoding=ISO-8859-1

Encoding database string values. If encoding=ISO-8859-1 is set, then the value will be encoded from base encode to ISO-8859-1.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
encodingTo=ISO-8859-8

Decoding database string values. If encoding is set to ISO-8859-1 and encodingTo is set to ISO-8859-8, then the result will be decoded ISO-8859-1 -> ISO-8859-8.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
modalityAggregate=true

Attempt to collect all modalities for every study in the search results.

false might improve performance in case of MySQL with corresponding configuration, however then only a single modality from some series will be visible.

• Example: com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
storage=fs1=c:\mnt\NAS1;fs2=c:\mnt\NAS2

Filesystem root directories, as configured in the LDAP tree via entries named “dcmStorageID”. The plugin is unable to read this configuration from LDAP at the moment.
4. Orthanc plugin

• Example:  `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].pacsVersion=5.7`

Two-component DCM4CHEE version number that defines what database fields are available. Namely, *Number of Frames* was added in 5.7 and this version is expected by default; if your PACS is older, you’ll need to adjust this parameter to avoid SQL errors.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].type=orthancPlugin`

Use this specific value of `type` when connecting to Orthanc.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].id=Orthanc`

Identifies the plugin/PACS pair in the search window drop-down menu. You can connect to multiple Orthanc instances by using different values of `id` and remaining options.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].searchApiEnabled=true`

Specifies if this plugin/PACS pair returns a study list for the Search window and provides study metadata.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].imageApiEnabled=true`

Specifies if this plugin/PACS pair retrieves image data.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].baseUrl=http://127.0.0.1:8042`

Base URL that points to the root of Orthanc Web API (`/tools/find`, `/instances` etc will be appended automatically).

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].username=UserName`

Username for Basic Authentication.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].password=UserPassword`

Password for Basic Authentication.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].dicomCacheDirectory=C:\DICOM`

*(optional)* A directory for caching images. Without this setting a file is downloaded anew every time and it’s not possible to examine its contents. Multiple instances of this plugin should have different directories in order to not mix up different versions of the file with the same Study/Series/SOP UIDs.

NOTE: only directories below `${com.softneta.meddream.tempDir}/cache` are cleaned automatically. A different path will require some third-party solution for free space management.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].strictSearchIsEnabled=true`

*(optional)* `true` enables exact matching by Patient ID in the Search window.

5. Query/Retrieve plugin

This is the equivalent of `$pacs = "DICOM"` in config.php.

• Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].type=QRPlugin`
Use this specific value of type when connecting to a PACS over the DICOM 3.0 protocol.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].id=QRPlugin`
  Identifies the plugin/PACS pair in the search window drop-down menu. Multiple instances of this plugin, however, are not recommended at the moment.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].searchApiEnabled=true`
  Specifies if this plugin/PACS pair returns a study list for the Search window and provides study metadata.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].imageApiEnabled=true`
  Specifies if this plugin/PACS pair retrieves image data.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].remoteAET=PACS`
  The AE Title of the remote machine.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].remoteHost=127.0.0.1`
  IP address or hostname of the remote machine.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].remotePort=104`
  Port number of the remote machine.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].localAET=MEDDREAM`
  The AE Title of the local DICOM C-STORE receiver.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].cacheDir=C:\DICOM`
  A directory for receiving images.
  
  **NOTE:** only directories below `${com.softneta.meddream.tempDir}/cache` are cleaned automatically. A different path will require some third-party solution for free space management.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].timeout=90000`
  DICOM file receive timeout in milliseconds.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].retryDelay=1000`
  DICOM file existence check period in milliseconds.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].strictSearchIsEnabled=true`
  true enables exact matching by Patient ID in the Search window.

- **Example:** `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].multipleModalitySupport=true`
  false will yield an error message when the user selects multiple modalities. Use this if the PACS does not support multiple modalities and search results confuse the users.
6. EDGE plugin

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].defaultStartDate=20181107`
  Default “from” date in YYYYMMDD format when empty in the user interface.

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].defaultEndDate=20181108`
  Default “to” date in YYYYMMDD format when empty in the user interface.

7. Aycan plugin

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].type=AycanPlugin`
  Type of plugin. Must be set to AycanPlugin.

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].id=AycanLocal`
  Identifier of the plugin.

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].searchApiEnabled=true`
  Default is false, must be set to true.

- Example: `com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].imageApiEnabled=true`
  Mandatory PacsGW property, should be set to true.
Default is false, must be set to true.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  url=jdbc:postgresql://127.0.0.1:5432/store
  
  Connection string.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  username=username
  
  User name for database connection.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  password=password
  
  Password for database connection.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  encoding=utf8
  
  Character set used for encoding/decoding database strings (e. g., latin1, cp1251, utf8).
  Default value is empty (no encoding/decoding).

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  mappedStorageLocation=/dicom/data/raid1=/mnt/NAS1;/dicom/data/raid2=/mnt/NAS2
  
  Maps study storage directories to some local equivalents. Format: “FROM1=TO1;FROM2=TO2;...”

8. Filesystem plugin

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  type=FileSystemPlugin
  
  Always set to FileSystemPlugin.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  id=File-System-Plugin
  
  Pacs storage ID name.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  searchApiEnabled=true
  
  Default is false, must be set to true.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  imageApiEnabled=true
  
  Default is false, must be set to true.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  rootDirectory=C:\dicoms\files
  
  Path to the base directory below which the DICOM files can be found.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  maxDepth=1
  
  The number of subdirectories traversed when collecting files. Minimal allowed value and default value is 0.

9. DCMSYS plugin

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  type=dcmsys
  
  Use this specific value of type when connecting to DCMSYS Router.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].
  id=default
Name of storage as defined in DCMSYS Administrator Site.

Multiple occurrences of type=dcmsys with different id is a standard way for supporting more than one storage.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].searchApiEnabled=true
  
  Specifies if this plugin/PACS pair returns a study list for the Search window.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].imageApiEnabled=true
  
  Specifies if this plugin/PACS pair retrieves image data.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].qidoUrl=https://127.0.0.1:8282/api/router/qido-rs
  
  Base URL for QIDO-RS requests used when searching for studies.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].wadoUrl=https://127.0.0.1:8282/api/router/wado-rs
  
  Base URL for WADO-RS requests used to access study metadata.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].dicomWadoUrl=https://127.0.0.1:8282/WADO
  
  Base URL for WADO-URI requests used to retrieve DICOM images.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].loginUrl=https://127.0.0.1:8282/api/login
  
  URL for the login endpoint.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].username=UserName
  
  Username for the login endpoint.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].password=UserPassword
  
  Password for the login endpoint.

- **Example:** com.softneta.meddream.pacs.configurations[PLUGIN_INDEX_NUMBER].dicomCacheDirectory=/MedDreamTemps/temp

  (optional) A directory for caching images. Without this setting a file is downloaded anew every time and it’s not possible to examine its contents. Multiple instances of this plugin should have different directories.

  **NOTE:** only directories below ${com.softneta.meddream.tempDir}/cache are cleaned automatically. A different path will require some third-party solution for free space management.

---

9. Image Access from hospital information system (HIS)

9.1. Generic HIS integration

HTTP POST or GET methods can be used to open a study in MedDream. The table below describes attribute types for study query.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>URI</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study UID</td>
<td>URL?study=STUDY_UID</td>
<td>Unsupported by File System</td>
</tr>
<tr>
<td>Patient ID</td>
<td>URL?patient=PATIENT_UID</td>
<td>Unsupported by File System</td>
</tr>
<tr>
<td>Patient ID and Accession Number</td>
<td>URL?patient=PATIENT_UID &amp;accnum=ACCESSION_NO</td>
<td>Supported only for PacsOne in external.sample-azimuth-pacsone.php</td>
</tr>
<tr>
<td>Series UID</td>
<td>URL?series=SERIES_UID</td>
<td>PacsOne and DCM4CHEE only, external.sample-rssdimed-*_.php required</td>
</tr>
<tr>
<td>File Name</td>
<td>URL?file=PATH_TO_FILE</td>
<td>File System only</td>
</tr>
<tr>
<td>*</td>
<td>-</td>
<td>The EDGE PACS is supported in interactive mode only (no HIS integration).</td>
</tr>
</tbody>
</table>

The most popular example:  

To enable study query from HIS, the following steps must be performed:

1. Among files external.sample-*_.php, find the one dedicated to your PACS and rename/copy it to external.php.

   `external.sample-FULL.php` is not recommended as it gets only compatibility fixes (no feature or security updates). It might be completely removed in the future. Please use a more specific example and email info@softneta.com in case the “FULL” example contains something important to your installation while the more specific one doesn’t.

2. Modify first lines of external.php that look similar to these:

   ```php
   define("SHOW_ENABLED", true); // true: HIS integration is enabled; false: disabled
   define("SHOW_DB", "dbname"); // database name
   define("SHOW_USER", "user"); // a database user
   define("SHOW_PASSWORD", "password"); // corresponding password
   ```

3. To add the study into the viewer opened in an existing browser tab, use addStudy.php instead of index.php:

   ```
   http://localhost/meddream/addStudy.php?study=1.2.392.200036.9107.500.110113&viewer=html&token=363b122c528f54df4a0446b6bab05515
   ```

   **NOTE:** the parameter “token” is optional by default. It becomes mandatory only with certain versions of external.php (external.sample-dcm4chee.php, external.sample-rssdimed-pacsone.php) and only if they configure a non-empty SHOW_SSO_ADDR.

When used this way, the viewer will contain multiple studies as if they were opened from Search without closing the browser tab containing the viewer.

Copy the following to an HTML file in the MedDream root directory, then update the URL near end of the file.

```html
<!DOCTYPE html>
<html lang="lt">
<head>
  <meta charset="utf-8">
  <title>MedDream</title>
</head>
(continues on next page)
```
9.2. Specification of thumbnails service

Provides access to thumbnails of study images via HTTP GET/POST request. One request lists images in the study and the second request fetches a single JPEG image.

IMPORTANT: Will reuse existing login information if MedDream is already open in the same browser (other tab or window). For standalone use, you need to configure external.php.

IMPORTANT: Both getImageList.php and getThumbnail.php are by default placed into the subdirectory “scripts” inaccessible from Web. If you need this functionality, then move or copy those files into the parent directory.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>MedDream root hyperlink without the query part (e.g., <a href="http://localhost/meddream/">http://localhost/meddream/</a>). It is OK to include index.php or home.php.</td>
</tr>
<tr>
<td>image_uid</td>
<td>SOP Instance UID</td>
</tr>
<tr>
<td>study_uid</td>
<td>Study Instance UID</td>
</tr>
<tr>
<td>image_size</td>
<td>Limit of image width and height. The image will be resized so that the larger side won’t exceed this value. Default value is 50, allowed range is 50…4320.</td>
</tr>
<tr>
<td>data_result</td>
<td>Choice of images to list: 0 - first image from the first series 1 - first image from every series 2 - all images from the study Default value is 0.</td>
</tr>
</tbody>
</table>

Request: list available thumbnails

URL/getImageList.php?study=study_uid&size=image_size&result=data_result

Returns a JSON-formatted array:

```json
{
  "studyUID": study_uid,
  "thumbnails": [
    {
      "imageUID": image_uid_1,
      "url": URL/getThumbnail.php?image=image_uid_1&size=image_size
    },
    ...,
    {
      "imageUID": image_uid_N,
      "url": URL/getThumbnail.php?image=image_uid_N&size=image_size
    }
  ]
}
```
In case of not found study, authentication failure or any other error, will return the HTTP 404 error with an empty Response Body.

**Request: get a thumbnail image**

URL/getThumbnail.php?image=image_uid&size=image_size

Returns the raw JPEG stream with an appropriate Content Type header.

In case of not found image, authentication failure or any other error, will return the HTTP 404 error with an empty Response Body.

### 9.3. Specification of Live Stream

Addresses of live streams can be defined in the file `liveStreamList.json` in the MedDream installation directory (for example, near the file `Routes.php`). The configured items will appear in MedDream search results as studies.

**IMPORTANT:** a live stream address must be valid and accessible via VLC player on every client machine. You must make sure that the address is accessible to the client: port is open, antivirus or firewall software isn’t blocking it, etc.

The file `liveStreamList.json` must contain valid JSON data.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>liveStream</td>
<td>Stream file block. Required.</td>
</tr>
<tr>
<td>id</td>
<td>Stream id. Short word. Required.</td>
</tr>
<tr>
<td>name</td>
<td>Stream short name.</td>
</tr>
<tr>
<td>modality</td>
<td>Set to “LIVE”.</td>
</tr>
<tr>
<td>description</td>
<td>Detailed description of the stream, a single line.</td>
</tr>
<tr>
<td>sourceAE</td>
<td>Full source address that can be played in the VLC player.</td>
</tr>
</tbody>
</table>

**Example**

```json
{
    "liveStream": [
    {
        "id": "1",
        "name": "Room #2",
        "modality": "LIVE",
        "description": "Operating room in 2nd floor\ndasdasd",
        "sourceAE": "rtsp://192.168.1.1:554/session0.mpg"
    },
    {
        "id": "2",
        "name": "Room #3",
        "modality": "LIVE",
        "description": "Operating room in 3rd floor",
        "sourceAE": "rtsp://192.168.1.2:554/session0.mpg"
    }
    ]
}
```

### 9.4. Further reading

More details can be found in a separate document, `quick_install-HIS_integration.txt`. 
## 10. Additional software

MedDream can be used in tandem with several external pieces of software that are listed below.

NOTICE: to get the required software, please contact support@softneta.com.

### 10.1. 3D service

#### 10.1.1. 3D server configuration

The XStream HDVR WebSDK runs with default system configuration (non-SSL through port **8088**) with the most common quality and performance settings. Default behavior is described below and is changed by modifying the file `foviaserver/config/configuration.xml`. The server must be restarted for any changes to take effect. The system configuration and default settings are described below:

```
<nodePort>8088</nodePort>
<useSSL>0</useSSL>
<jpegInteractive>50</jpegInteractive>
<jpegFinal>95</jpegFinal>
```

#### 10.1.2. Network settings (ports and SSL)

`nodePort` and `useSSL` above specify the network protocol and port to be used. Separate port configuration settings are used for SSL and non-SSL, although only one can be used at one time.

The port can be changed to any valid unused port, such as 9123. After restarting your server, validate that the test app “helloFovia” still works correctly after the corresponding reconfiguration (<FoviaServerIPaddress>. ...:9123/version...).

To enable SSL, set `useSSL` to 1 and restart your server. You can test the HTTPS protocol on the URL `https://localhost:8088/version` which uses the supplied test certificate. In a production environment, replace the supplied demo files `server.crt` and `server.key` in the directory `foviaserver/ssl` with your company’s official certificate.

#### 10.1.3. Quality / Performance

Default values for quality and performance are explained below. Care must be taken when changing these settings since they affect all users on the system. There are two types of quality / performance metrics in the system: those that relate to network, and those that relate to the actual rendering (and ray casting).

The network quality and performance trade off affects how much bandwidth is required for transferring images over the network, and specifically the amount of JPEG compression for interactive and final quality images. This is largely independent of rendering quality that is described below.

There are two levels of JPEG compression: `jpegInteractive` is set to 50/100 and `jpegFinal` is set to 95/100. Typically during mouse interactions, many frames of quality `jpegInteractive` are generated (such as during the render() call during a mouseMove operation). These are quick rendering operations with more aggressively compressed JPEG images. On mouseUp, a `renderFinal()` operation is triggered and results in a frame of quality `jpegFinal`. The rendering engine will take longer to generate higher quality images, and JPEGs with less aggressive compression will be used for these.

The server version and configuration settings are displayed at `http://localhost:8088/version`.

The 3D service creates a 3D representation of the study. In order to use the 3D service the MedDream license must have the “3D” module and the following files have to be edited.
10.1.3.1. system.json

In this file, update the parameter `threeDimensionServiceIntegration`:

```
{
    "threeDimensionServiceIntegration": [{
        "serverIP": "127.0.0.1",
        "port": "8088",
        "protocol": "http:",
        "pathPrefix": null,
        "pathPrefixReplaceTo": null
    }]
}
```

- **serverIP**
  Address of the server where the 3D service is installed. Default value: “127.0.0.1”.

- **port**
  The set port of the 3D service. Default value: “8088”.

- **protocol**
  The transfer protocol used by the server. Possible values: http, https.

- **pathPrefix**
  Used if the 3D server is on a different machine than the PACS and the latter keeps DICOM files locally. Sets the base path of the DICOM file storage folders (like D:\DICOM). Default value: null.

- **pathPrefixReplaceTo**
  A local equivalent of pathPrefix. At least this contains the path on the 3D server machine where DICOM file storage shared on the PACS machine is mounted (like Z:\DICOM). Default value: null.

10.1.3.2. Apache configuration file (httpd.conf)

If outside connections to the 3D service are not allowed, then the Apache configuration needs the following lines:

```
<IfModule proxy_module>
  <IfModule proxy_http_module>
    RewriteEngine On
    RewriteCond %{REQUEST_URI} ^/socket.io [NC]
    RewriteCond %{QUERY_STRING} transport=websocket [NC]
    RewriteRule /(.*[^/]*) ws://{3D_SERVICE_IP}:{3D_SERVICE_PORT}/$1 [P,L]
    ProxyPass /socket.io http://{3D_SERVICE_IP}:{3D_SERVICE_PORT}/socket.io
    ProxyPassReverse /socket.io http://{3D_SERVICE_IP}:{3D_SERVICE_PORT}/socket.io
  </IfModule>
</IfModule>
```

**NOTE:** When SSL is used, change ws to wss and http to https.

10.2. Browser plugin

A plugin for the Chrome browser that expands a newly opened tab across several monitors. The URL is scanned for a entered keyword and if a match is found, a new window is created and expanded across the selected monitors. The plugin can be added using the Chrome extension tab.

**NOTICE:** to get the plugin, please contact support@softneta.com
10.3. OpenAM verification service

OpenAM is an open source access management and entitlements server platform. This replaces the standard MedDream verification service. Installation is explained in the next chapter.

NOTICE: to get a prepared OpenAM package, please contact support@softneta.com

11. MedDream and OAuth

MedDream 6.0 and later versions add a possibility to use OAuth for authentication and user rights management. Implementation is based on OpenAM.

The file external-oauth2.php provides client-side integration. It is based on external.sample-pacsone.php and is therefore to be used with $pacs = 'PacsOne' (config.php). However if you don’t need HIS integration and authentication is enough, then this is a working authentication solution for configurations like $pacs = 'DICOM'.

11.1. Installation

Run OpenAM server on embedded Tomcat from the installation folder (OPENAM_HOME):

```
java -jar openam-server.jar
```

It will start server on default 8090 port. To run on specific port for example 8088:

```
java -jar openam-server.jar -port 8088
```

**OpenAM server**

**http://host:port/openam**

**Initial setup**

Make sure OpenAM server is started. During configuration enter:

**OpenAM server url** - http://host:port


**Windows**

Run script:

```
OPENAM_HOME\scripts\config.bat
```

**Linux**

Set access rules:

```
chomd a+x OPENAM_HOME/scripts/*.sh chomd 400 OPENAM_HOME/tools/admin/pwd.txt
```

Run script:

```
OPENAM_HOME/scripts/config.sh
```

**Remove configuration**

Stop OpenAM server

**Windows**

Run script:

```
OPENAM_HOME\scripts\remove-config.bat
```

**Linux**
Run script:
OPENAM_HOME/scripts/remove-config.sh

**PacsOne users import**

"mysql" must be accessible from any location.

**Windows**

Run users extraction script:
OPENAM_HOME\scripts\extract-users.bat <users export file> <db host> <db user> <db password> <db name>
Real example:
OPENAM_HOME\scripts\extract-users.bat OPENAM_HOME\scripts\users 192.168.11.10 root password dev
Run users import script:
OPENAM_HOME\scripts\import-users.bat <users export file>
Real example:
OPENAM_HOME\scripts\import-users.bat OPENAM_HOME\scripts\users

**Linux**

Run users extraction script:
OPENAM_HOME/scripts/extract-users.sh <users export file> <db host> <db user> <db password> <db name>
Real example:
OPENAM_HOME/scripts/extract-users.sh OPENAM_HOME/scripts/users 192.168.11.10 root password dev
Run users import script:
OPENAM_HOME/scripts/import-users.sh <users export file>
Real example:
OPENAM_HOME/scripts/import-users.sh OPENAM_HOME/scripts/users

12. MedDream and external proxies

After a successful login index.php (or home.php) redirects to itself in order to store credentials in the PHP session. By default the target URL depends on $_SERVER['HTTPS'] and $_SERVER['HTTP_HOST'].

If MedDream is served by a machine inside a LAN, which is exposed to Internet by a proxy server only, then the browser will receive a URL containing the LAN-specific server name from $_SERVER['HTTP_HOST'], and likely won’t be able to open it. In this case, set DEFAULT_USE_PROXYED_HOST = true in meddream/WebTarget.php and make sure the proxy server adds the X-Forwarded-Host HTTP header.

Some installations enforce HTTPS only for the “outside” connection: the browser uses HTTPS when connecting to the proxy, while the proxy uses HTTP when connecting to MedDream. Usually this is not a problem as the redirect to HTTP is converted to a HTTPS redirect. If the proxy doesn’t handle this (HTTP isn’t working at all), then you can set DEFAULT_USE_PROXYED_PROTO = true in meddream/WebTarget.php and add X-Forwarded-Proto header in the proxy server.

In a similar fashion DEFAULT_USE_PROXYED_PORT = true together with the X-Forwarded-Port header could be used to override the port. As $_SERVER['HTTP_HOST'] normally includes the port, you might need to override both port and host.
13. Security considerations

RATIONALE: It is an old dilemma of using “security through obscurity”. As a matter of fact, most MedDream installations tend to have a few security holes due to non-paranoid Web administrators. The very publication of this knowledge makes every old/unmaintained MedDream installation an easy target. But, we must draw a line one day so that at least new installations are secure.

13.1. Search engines

If MedDream is exposed to the Internet, your installation might be found simply by searching for “MedDream” or “Softneta”. (Rebranding is possible but most customers don’t need it.) Afterwards an attacker can check for typical misconfigurations as explained later in this document.

Solution: set up the robots.txt file.

This file is always hosted as /robots.txt (at the website root). If MedDream is also hosted there, the following robots.txt will be sufficient:

```
User-agent: *
Disallow: /

User-agent: AdsBot-Google
Disallow: /
```

A recommended read is http://www.robotstxt.org/robotstxt.html.

13.2. The subdirectory ‘log’

This subdirectory may contain files php-YYYYMMDD.log with sensitive patient data. Logs are not enabled by default, but might be needed during troubleshooting which sometimes takes days. Also one might forget to remove them (or disable logging) afterwards.

Solution: disable Web access to this directory altogether.

When using Apache, this is achieved by the following .htaccess file in this directory:

```
Order deny,allow
Deny from all
```

Apache 2.4+ requires a different syntax:

```
Require all denied
```

Reminder: If you need a restrictive AllowOverride directive on a related directory, its compatible values are “Limit Options” (first one allows Order/Deny, and the second one is for Options -Indexes in the next chapter). The equivalent for 2.4+ is “AuthConfig Options”. Furthermore, MedDream requires “FileInfo” since 6.1 so you must add this one when changing AllowOverride All to a more detailed subset.

Since v7.1 (and corresponding hotfixes for older versions), MedDream already contains a universal .htaccess file in this directory.

13.3. The subdirectory ‘temp’

The subdirectory “temp” is accessed via Web when viewing DICOM MPEG2 (Transfer Syntax UID = 1.2.840.10008.1.2.4.100) files, and when downloading an .ISO/.burn file after Export function finishes. It might
contain temporary files and subdirectories with randomized names and DICOM/JPG/FLV data inside; every such file might reveal sensitive patient information. Some of these files are left undeleted when MedDream crashes, and some are just impossible to delete on time. Scripts deleteTemp.* can be regularly called by the operating system’s scheduler to minimize amount of rubbish, however the latter will not disappear completely.

**Solution: disable Web listing of this directory.**

When using Apache, this is achieved by the following .htaccess file in this directory:

```
Options -Indexes
```

Alternatively, the entire MedDream directory can be denied listing in the `<Directory>` configuration block.

Furthermore, if you do not intend to view MPEG2 or use the Export function (or MedDream doesn’t support them in your configuration), then simply disable Web access to this directory altogether, as with “log” subdirectory above.

Since v7.0 (and corresponding hotfixes for older versions), MedDream already contains `Options -Indexes` in meddream/.htaccess which affects subdirectories as well.

### 13.4. printVersions.php

This file returns versions of major MedDream components **without requiring authentication**. It is not needed by MedDream itself and is rather dedicated to help investigations by service staff and automated deployments.

If you believe that versions might suggest known vulnerabilities and therefore speed up their search, then at least rename this file to something obscure.

### 13.5. DCM4CHEE 2.x

Official installation instructions offer some defaults that sometimes are left unchanged.

The default database name for 2.x is “pacsdb”. You also might be tempted to call it “dcm4chee”. For a knowledgeable attacker, both provide a clear suggestion which PACS is there and what to try next.

After a mindless installation there also will be some default user accounts:

- a database user “pacs” with password “pacs”. Perfectly suitable for MedDream;
- an internal user “admin” with password “admin” – suitable for MedDream and DCM4CHEE’s web interface;
- an internal user “user” with password “user” – suitable for MedDream and DCM4CHEE’s web interface.

Any of these accounts can be used to access sensitive patient data. We suggest to change passwords for all three, as soon as possible after the installation. The first one is used by DCM4CHEE to connect to the database, therefore you will also need to update the file server/default/deploy/pacs-*-ds.xml (name depends on database used).

### 13.6. The subdirectory ‘scripts’

This subdirectory contain files with background jobs and related logs.

**Solution: disable Web access to this directory altogether.**

When using Apache, this is achieved by the following .htaccess file in this directory:

```
Order deny,allow
Deny from all
```

(Apache 2.2) or
(Apache 2.4+).

Since v7.1 (and corresponding hotfixes for older versions), MedDream already contains a universal .htaccess file in this directory.

### 13.7. prepared.php

This file (since 6.1) allows to manage failed preparation jobs, examine DICOM tags and even download entire objects. Access is protected by a password (no user name) that is stored unencrypted in a configuration file. Until 7.1, the default password was known to everyone who downloads the MedDream installation archive.

**Solution:** change the password as soon as possible after the installation.

File meddream/scripts/prep/config.php:

```php
define('WEB_PASSWORD', 'the new password'); // used by prepared.php, without user name */
```

Since v7.1 (and corresponding hotfixes for older versions), MedDream does not accept neither an empty password which is now the default, nor the old non-empty “compromised” value.

### 13.8. Embedding into IFRAME etc

Availability to embed MedDream into FRAME, IFRAME or OBJECT containers allows untrusted websites to use MedDream in fraudulent manner (“clickjacking”). Since v6.2.1 this is restricted by default: only the server that hosts MedDream is allowed to wrap it with an additional container.

This is implemented by the X-Frame-Options header in AuthDB.php:

```php
public function addXFrameOptionsHdr()
{
    /* alternatively you can configure the header globally in the webserver and comment this line out */
    header('X-Frame-Options: sameorigin');
}
```

You can comment out the entire line to remove the restriction, or change `sameorigin` to `deny` in order to disallow any embedding.

**It is recommended to add the header to webserver's configuration instead.** In that case comment out the line above and add the following to the website configuration:

```
Header always set X-Frame-Options "sameorigin"
```

Other possible values of the header and their support in browsers can be found [here](#).

What if multiple hosts should be allowed to embed MedDream, while keeping others restricted? The mentioned place of AuthDB.php, or the webserver’s configuration, can be used to add the Content-Security-Policy header. Internet Explorer will need another header, `X-Content-Security-Policy: sandbox`.

Using the java based core it is possible to set IFRAME permissions in the application.properties file. Valid values: `NONE,DENY,SAMEORIGIN,ALLOW-FROM`. When `NONE` is used, then header is not set and when `ALLOW-FROM` is used, then valid host must be set in parameter `security.frameOptionsWhitelist`. Address entries are separated using spaces.
13.9. saveImage.php/saveSeries.php and HIS integration

Due to a design bug, UID validation makes saveImage.php and saveSeries.php nonfunctional when a study is opened from HIS integration and the Java backend is used (usePacsGwDirectly=true by default since 7.1) together with PACS plugins (need to be configured in application.properties). As this is the recommended configuration, validation is disabled by default in those two files.

If PACS plugins are not used (regardless of usePacsGwDirectly), or you aren’t using the Export function at all (for example, even the button is hidden), then we encourage to turn the validation back on. In both files, this line near the beginning needs to have `true` instead of `false`:

```php
use Softneta\MedDream\Core\ActionFilter;
use Softneta\MedDream\Core\RetrieveStudy;
```

13.10. php.ini

Some PHP settings might make the installation less secure. On the other hand, some other settings must be “less secure” for MedDream to work, compared to popular expectations. This chapter lists most important recommendations and caveats.

**Error handling**

- `expose_php = Off`
  
  The value “Off” hides the version of PHP from webserver response headers or error messages and therefore makes it more difficult to scan for known vulnerabilities of PHP itself.

- `error_reporting = E_ALL`
  
  “E_ALL” is recommended for production. However during installation/troubleshooting of any server, including a production one, a more detailed choice like “E_ALL | E_STRICT” might be needed. One could have two versions of this line and comment the inactive one out.

- `display_errors = Off`

- `display_startup_errors = Off`
  
  Use “Off” in production for both, or else error messages visible to the end user might reveal paths to files and other sensitive information.

- `track_errors = Off`
  
  Can remain disabled as MedDream doesn’t use `$php_errormsg`.

- `html_errors = Off`
  
  “On” is not needed and, due to `display_errors=Off`, has no effect anyway.

- `log_errors = On`

- `error_log = /valid_path/PHP-logs/php_error.log`
Logging to a file is preferred over STDERR. The latter means Apache’s error_log where PHP messages will be intermixed with Apache messages and therefore harder to read.

- `ignore_repeated_errors = Off`
  “On” is OK only as a temporary measure if there are lots of identical errors, they are filling up the log and nothing else can be done about that right now.

**General settings**

- `doc_root = /MedDream/directory`
  The option is useful only in CGI/FastCGI mode unsupported by MedDream. You can still safely specify the MedDream installation directory or some parent part of that path, if a dumb security scanner insists.

- `open_basedir =`
  For simplicity, leave this empty. MedDream must be able to access not only its own files below the installation directory, but also DICOM files that often are on a totally different mount point (or drive letter). A directory tree that includes everything might be just not worth to implement, and increases risk of exposing DICOM files to Internet.

- `include_path = .:/path/PHP-pear/`
  Some parts of MedDream still depend on “.” in this parameter so don’t remove it. Code from PEAR is not used by MedDream, but might be used by other software if hosted on the same server.

- `extension_dir = /path/PHP-extensions/`
  Use a full path here to avoid surprises like “Apache loads php_meddream just fine but scripts started from cron or Task Scheduler do not".

- `mime_magic.magicfile = /path/PHP-magic.mime`
  Use a full path. The Reporting function needs a correct guess of MIME Type during attachment upload, in order to correctly indicate it later during download.

- `allow_url_fopen = On`
  As of 7.1.1, MedDream still depends on the value “On” for communication between some components (not everything has been ported to cURL).

- `allow_url_include = Off`
  “Off“ for both is safer and doesn’t interfere with any MedDream functionality.

- `variables_order = "GPCS"
  “GPCS” corresponds to four superglobals on which MedDream depends; there is no need to include “E”.

- `session.gc_maxlifetime = 600`
  A smaller value will log the user off faster if the browser is closed, the computer is suspended or the Internet connection is cut. 10 minutes is a sane default, a safe minimum is approx. 3 minutes, and the legacy default is 1440 (24 minutes).

**File uploads**

- `file_uploads = On`
  Needed only for uploading attachments in the Reporting function, otherwise “Off” is a safer choice.

- `upload_tmp_dir = /path/PHP-uploads/`
Use a full path. A separate directory might make cleaning leftover files easier.

- upload_max_filesize = 10M
- post_max_size = 40M

Attachments larger than upload_max_filesize will fail to upload. To account for overhead, ensure post_max_size > 4*upload_max_filesize and memory_limit > 16*post_max size.

- max_file_uploads = 1

MedDream always uploads a single attachment per request so “1” is a safe choice.

Executable handling

- enable_dl = Off

The MedDream PHP extension is expected to load automatically. The dl() function is not used.

- disable_functions = system, shell_exec, passthru, phpinfo, show_source, highlight_file, fopen_with_path, dbmopen, dbase_open, putenv, chdir, filepro, filepro_rowcount, filepro_retrieve, posix_mkfifo

These functions are generally considered dangerous and MedDream does not use them. This list is a safe choice.

In turn, some functionality or configurations of MedDream require other “dangerous” functions, namely: exec, popen, proc_open, move_uploaded_file, mkdir, rmdir, chmod, rename.

Session handling

- session.save_path = /path/PHP-session/

Use an absolute path. This will ensure that both Apache-integrated and CLI versions of PHP keep session files in the same location.

- session.name = PHPSESSID

Do not change: this value is hardcoded in Java and used when calling PHP from some Java endpoints for authentication check.

- session.auto_start = Off

MedDream is developed with this “Off” in mind. Not only is “On” less safe, but also MedDream was not tested with it.

- session.use_trans_sid = 0
- session.use_cookies = 1
- session.use_only_cookies = 1

MedDream explicitly expects some cookies, do not disable them at client side. Consequently, use_trans_sid=1 and use_only_cookies=0 also have no sense.

- session.cookie_domain = my.website.tld

If MedDream is always accessed as “my.website.tld”, then this domain is worth specifying in session.cookie_domain.

If, however, a hospital has a habit of accessing MedDream via a different name (or even IP address) from its intranet than from Internet, then this setting might need to remain empty.

- session.use_strict_mode = 1

“1” or “On” is more secure. There is no effect on MedDream functionality in any case.

- session.cookie_lifetime = 0

Zero is the legacy value (the session exists while the browser is not closed) and end users are accustomed to it.
A nonzero value would impose a hard limit after which the end user will need to log in again, even if actively using MedDream at the moment (by the way, MedDream does not handle this situation gracefully). Furthermore, if the browser is closed earlier without logging off, then the session cookie will remain in the persistent storage until the time expires, and could be used for impersonation.

- `session.cookie_secure = 1`
  
  If MedDream is accessed only via HTTPS (plain HTTP is undesirable), then set this to “1”.

- `session.cookie httponly = 1`
  
  MedDream assigns “1” internally to compensate against absence of this value in php.ini. However you should still update php.ini as a safeguard, especially for other software if hosted on the same server.

- `session.cookie_samesite = Strict`
  
  This option was added to PHP 7.3 not supported by MedDream yet. You can still add it (will be ignored by older PHP versions) as a reminder for a future upgrade.

- `session.cache_expire = 180`

- `session.cache_limiter = nocache`
  
  These are legacy defaults with which MedDream is still tested. In fact, cache_expire is ignored when cache_limiter=nocache, and this will disallow caching in any intermediate Web proxies. MedDream controls caching of some functionality-related backend responses on its own, and in these cases the global caching settings won’t have any effect.

- `session.sid_length = 256 # PHP 7.1+`

- `session.sid_bits_per_character = 6 # PHP 7.2+`

- `session.hash_function = 1 # PHP 7.0-7.1`

- `session.hash_bits_per_character = 6 # PHP 7.0-7.1`

  The above values are recommended and are understood by newer PHP versions.

- `session.referer_check = http://server/meddream/path`

  Entering MedDream root URL (like “http://server/meddream” or “http://server/”) might guard against some attacks. This is obviously suitable only if MedDream is always accessed by the same URL (for example, regardless if it’s done from Internet or from intranet).

**Less obvious settings**

- `memory_limit = 128M`

  As per popular recommendations (targeted to any kind of PHP applications), this value should be as small as possible, like “50M”, for early termination of inefficient code.

  But, some DICOM files might require a huge amount of memory due to being large or complex, so this setting becomes important when using the PHP backend ("usePacsGwDirectly":false in system.json). The typical default of “128M” sometimes still results in out of memory errors, therefore you’ll be forced to increase it.

- `max_execution_time = 30`

  Similarly to memory_limit, the legacy default of 30 seconds might be insufficient. On the other hand, MedDream controls this setting internally whenever a long-running operation is expected, so a need to increase max_execution_time can, in some cases, be treated as a MedDream bug.

- `report_memleaks = On`

  “On” might help to detect memory leaks earlier.
14. Localization

By default MedDream is available in English, Lithuanian and Russian; user documentation is only in English. Softneta is not responsible for not verified translations made by licensed users. Please note that Medical Device certificates are not valid for not verified translations.

You can specify any available language or their combination in meddream/config.php. For example,

```bash
--- a/config.php 2018-10-12 11:30:52.663417900 +0300
@@ -120,7 +120,7 @@
     Example: "en,lt,ru"
 */
- $languages= "en";
+ $languages= "en,lt,ru";

/* default login credentials (for demonstration purposes etc)

The first language listed here will be offered by default on user’s computer.

15. Rebranding

MedDream can be rebranded and use custom branding information (e.g., logo, product name, contact details). Important things to consider:

- By default the software license does not allow rebranding of the About dialog. Please contact info@softneta.com for an updated license that is fully compatible with rebranding.
- The license can also define software as a module and, as a result, hide the following parts of the About dialog: “Manufactured by:”, “Medical device class:”, “ID of the notified body:”, “FDA Cleared:”, the FDA and CE labels.
- Rebranded software does not check for recent versions; the version displayed in About dialog is always considered up to date.
- If the login form tells that rebranding_configuration.json is not valid, you should check its format with some JSON editor or online resource (for example, http://jsonlint.com/) and also confirm that the “site” attribute begins with “http”.

Configuration and related files must be located in the directory APACHE_HTDOCS_DIR/meddream/rebranding/.

The main configuration file is `rebranding_configuration.json`. It must contain a valid JSON object without comments. The file must be encoded in UTF-8.

JSON object attributes:

- `isRebranded`
  - Possible values: true, false
  - Default value: false
  - This attribute is mandatory. It allows to turn on/off rebranding without changing a lot of other attributes. If the configuration file is missing, or attributes are empty, then corresponding images and contact information are not displayed.
  - Example:
    "isRebranded":true
• **productName**
  Possible values: any text
  Replaces the text “MedDream” in the bottom left corner of the login form, the “Product:” part of the About dialog, the title of the browser tab.
  Example:
  ```json
  "productName":"MedDream"
  ```

• **productVersion**
  Possible values: any text
  Replaces the version in the bottom left corner of the login form, and the “Version:” part of the About dialog.
  Example:
  ```json
  "productVersion":"1.1.1"
  ```

• **copyright**
  Possible values: any text
  Replaces the text “Copyright 2019 © Softneta” in the bottom left corner of the login form, and below search results.
  Example:
  ```json
  "copyright":"Copyright 2019 © Softneta"
  ```

• **companyLogoFile**
  Possible values: name (with extension) of an image file in the ./rebranding subdirectory
  Image format: PNG, JPG
  Replaces the company logo in the About dialog.
  Size of the default image is 134 x 28. The maximum width is 600 pixels, anything past this is clipped. Image height is not limited, however it will cause some repositioning of the dialog contents; a too high image might move the buttons outside the screen so the dialog becomes unusable. Make sure to check your customization on various client systems (different display DPI, etc) from where MedDream will be opened.
  Example:
  ```json
  "companyLogoFile":"companylogo.jpg"
  ```
  (refers to file `APACHE_HTDOCS_DIR/meddream/rebranding/companylogo.jpg`)

• **productLogoFile**
  Possible values: name (with extension) of an image file in the ./rebranding subdirectory
  Image format: PNG, JPG
  Replaces the product logo in viewer’s top left corner.
  Size of default image is 167 x 37. The height can be up to 51 px, anything more is clipped. If the width exceeds 224 px, the thumbnail stripe will move to right. When your picture is near these limits, it might need checking on various client systems to ensure consistent behavior.
  Example:
  ```json
  "productLogoFile":"productlogo.jpg"
  ```
  (refers to file `APACHE_HTDOCS_DIR/meddream/rebranding/productlogo.jpg`)

• **loginLogoFile**
Possible values: name (with extension) of an image file in the ./rebranding subdirectory
Image format: PNG, JPG
Replaces the picture above the login form.
The width should be 290 px exactly (see `.content` in assets/css/style.css) as the image is left-aligned in a container having this width.
The height is not limited. Of course the entire login form might not fit on smaller screens, especially in mobile devices, due to excessive height.
Example:

```
"loginLogoFile":"loginlogo.jpg"
```
(refer to file APACHE_HTDOCS_DIR/meddream/rebranding/loginlogo.jpg)

• companyName
  Possible values: any text, preferably single line.
  Replaces the 1st line under “CONTACTS:” in the About dialog.

• streetAddr
  Possible values: any text, preferably single line.
  Replaces the 2nd line under “CONTACTS:” in the About dialog.

• postCode, city
  Possible values: any text, preferably single line.
  A combination of these (postCode + , + city) replace the 3rd line under “CONTACTS:” in the About dialog.

• country
  Possible values: any text, preferably single line.
  Replaces the 4th line under “CONTACTS:” in the About dialog.

• email
  Possible values: any text, preferably single line.
  The 5th line under “CONTACTS:” in the About dialog will display this text. A prefix “E-mail: ” is always added to the displayed text. A prefix “mailto:” is always added to the underlying address.
  If this line is clicked, MedDream opens the default email client and offers to compose an email to this recipient.

• site
  Possible values: any text that begins with “http” (you can use “https”, too)
  The 6th line under “CONTACTS:” in the About dialog will display this text. The same text is always added as the underlying address.
  If this line is clicked, MedDream opens a new browser tab and navigates to this address.

• defaultTheme (HTML Viewer only)
  Possible values: red (default), blue, green NOTICE: default theme can not be applied for the login page.
  Example:

```
"defaultTheme":"blue"
```
A typical “all included” example of rebranding_configuration.json:
Some window names, namely, “MedDream search”, “MedDream viewer”, “MedDream Settings” visible on the tab header, currently can’t be rebranded this way. The recommended workaround is to update the translation file. For example, the English translation meddream\locales\en\translation.7.1.0.json for MedDream 7.1.0 could be changed as follows:

```json
--- translation.7.1.0.json.ORIG 2019-02-28 17:04:55.677247300 +0200
+++ translation.7.1.0.json 2019-02-28 17:12:45.988251300 +0200
@@ -21,7 +21,7 @@
     "DateTime": "Date Time",
     "ReceivedOn": "Received On",
     "SourceAE": "Source AE",
-    "Tittle": "MedDream search",
+    "Tittle": "MyProduct search",
     "About": "About",
     "LicenseAgreement": "License Agreement",
     "MoreThan": "Filtered more than 1000 records!",
@@ -72,7 +72,7 @@
     "Close": "Close",
     "Apply": "Apply",
     "Cancel": "Cancel",
-    "Tittle": "MedDream viewer",
+    "Tittle": "MyProduct viewer",
     "date": "Date:",
     "CloseStudy": "Close study",
     "CloseStudies": "Close studies",
@@ -239,7 +239,7 @@
     "notAvailable": "Share Dicom library not available when license do not allow
         version updates."
 },
-    "settings": {
+    "settings": {
        "Tittle": "MedDream Settings",
        "Tittle": "MyProduct Settings",
        "On": "Enable",
        "Off": "Disable",
        "Rows": "Rows:",
```
16. Troubleshooting

16.1. Log files of the legacy backend

The directory \texttt{APACHE_HTDOCS_DIR/meddream/log} contains two kinds of log files:

- \texttt{php-YYYYMMDD.log} – log messages from PHP-based core
- \texttt{YYYYMMDD-HHIISS-PPPPP.log} – log messages from the \texttt{php_meddream} module

By default this directory also contains the file \texttt{enabled} that is ignored; it will come into effect after renaming to \texttt{enabled} (without the underscore).

The file \texttt{enabled} configures the log level, specified by a single digit 0 to 4 (more levels might be added in the future):

- 0 – off
- 1 – errors
- 2 – warnings
- 3 – information
- 4 – debug

It is recommended to leave the value “1” in new installations. Currently, however, messages from the lower levels might have not enough context information to be usable.

After changing the level, it is necessary to restart Apache so that \texttt{php_meddream} also reacts to the change. Otherwise only \texttt{php-YYYYMMDD.log} is affected.

16.2. Audit logs

Starting from 6.2, the directory \texttt{APACHE_HTDOCS_DIR/meddream/log} by default also contains files \texttt{audit-YYYYMMDD.log}. Their main purpose is to log access to specific backend endpoints for security auditing. But, in some cases their contents might provide additional context for messages from \texttt{php-YYYYMMDD.log}.

Please see quick_install-HIPAA_audit_log.txt for details.

16.3. Log files of the Java-based core

In the directory of the Java application, the file \texttt{mdjavacore.log} will be created.

To control the overall logging level, add the following line to \texttt{application.properties}:

\begin{verbatim}
logging.level.root=DEBUG
\end{verbatim}

Alternatives to “DEBUG” are INFO, ERROR etc as per Log4j specification.

The existing setting \texttt{logging.level.com.softneta} refers only to internal MedDream logic and in some cases might be insufficient even at DEBUG level.

16.4. php_meddream error codes

Earlier the frontend used to display raw error codes from \texttt{php_meddream}, for example, “Parse error 4”. Those codes might still slip through; normally they will be visible only in the logs.

1, “Internal error”
Definitely a bug in php_meddream.

2, “Invalid parameter(s)”
   A bug in the caller of php_meddream (an earlier call provided enough information to prevent this).

3, “Not enough memory”
   You can try to increase memory_limit in php.ini. Note that 32-bit versions of Apache/PHP are limited to 2 GB anyway.

4, “Filesystem-related problem”
   Failed to open a file for reading (file or subdirectory missing?) or for writing (permissions? write lock?).
   Failed to read from a file (disk error?) or write to a file (full disk? disk error?).

5, “Invalid data or configuration”
   Wrong or unsupported value of a DICOM tag in the file.
   Length problems: a critical tag too long, too short or missing; entire file is empty or ends unexpectedly.

6, “Operating system-related problem”
   Failed to run a helper binary (mdc.exe on Windows, or FFmpeg).
   Windows only: mdc.exe reports various cases of non-success, including an unhandled exception.

9, “Decompressor precision too low, retry with a higher one”
   Can only happen with JPEGs that have jpeg_decompressstruct.data_precision > 16 (would be unsupported anyway, and we don’t have a real example).

10, “Image format not supported”
   SOP Class (or Samples Per Pixel, Bits Stored, Photometric Interpretation, etc) is unsupported. “SOP Class” also refers to known non-image compound objects.
   Transfer Syntax is unsupported and meddream.parser_choice disallows fallback to GDCM.

16.5. Browser download path selection

16.5.1. Google Chrome

You can choose a location on your computer where downloads should be saved by default or pick a specific destination for each download.

- On your computer, open Chrome.
- At the top right, click More and then Settings.
- At the bottom, click Advanced.
- Under the “Downloads” section, adjust your download settings:
  - To change the default download location, click Change and select where you’d like your files to be saved.
  - If you’d rather choose a specific location for each download, check the box next to “Ask where to save each file before downloading”.

If you didn’t change your default download location, then Google Chrome downloads files to the following locations:

- Windows 10, 8, 7 and Vista: Users<username>Downloads
- Mac: /Users/<username>/Downloads
• **Linux**: `/home/<username>/Downloads`

### 16.5.2. Mozilla Firefox

**Note:** Changing the location of your downloads affects **all** downloaded files in this Web browser.

- Click the menu icon on the top right corner of the browser.
- Click **Preferences**.
- Click **General**.
- Click **Choose**… next to Save files to.

### 16.5.3. Microsoft Edge

- Open Microsoft Edge.
- Select **Settings and more** -> **Settings**.
- Under **Downloads**, select **Change**.
- In the dialog box, select a new location for your downloaded files.

### 16.5.4. Safari

To change the default download location of your Safari browser:

- Click on the **“Edit Menu”** -> **Preferences** -> **General tab**
- Locate the **“Save downloaded files to”** section, Click on **“Downloads”** > **“Other”**…
- Browse and indicate your new download location.

### 16.5.5. Internet Explorer 11

- click on the **gear icon** (top right corner of the window), and choose **“View Downloads”** from the menu
- click on the **Options** link in the bottom left corner
- click on the **Browse** button and pick another folder. Click **OK** once done.

### 16.6. Symptom: on Linux the loading pauses just after logging in, possibly with wrong webpage colors

On some Linux installations, the Java backend might pause for up to several minutes when a new concurrent connection is made. Immediately after the login, the user sees either an empty dark page with a progress bar spinning at the top, or an empty white page, or a white page with a black-and-white DEMO message.

The cause: the default configuration of OpenJDK suggests reading from `/dev/random` that can block due to exhausted entropy pool. Some systems are refilling the pool more slowly so this problem becomes more common.

**Solution 1:** add `-Djava.security.egd=file:/dev/./urandom` to the command that starts the Java backend. Refer to the **Running as a service** chapter for hints where this command might be located.

It is recommended to stop the service, then start the updated command manually and verify the effect. You might want to force entropy exhaustion so that the problem appears as soon as possible; use `dd if=/dev/random of=/dev/null bs=512 count=1` for that (on a different terminal as it will block, too).
**Solution 2:** the above might still not help with some OpenJDK builds, therefore you’ll need to update system-wide configuration in the java.security file (traditionally it’s `$JAVA_PATH/jre/lib/security/java.security`):

```
securerandom.source=file:/dev/urandom
```