PalMA Installation Instructions

Source: https://github.com/UB-Mannheim/PalMA/blob/palma-redesign/INSTALL.md

Introduction

You can install PalMA manually with the descriptions provided in this document or you can run the **experimental installation script** we provide under `scripts/install_palma.sh` to do most of the work for you. You can call it like e.g. like so:

```
install_palma.sh install "/var/www/html" standard "https://www.your-institution.org/link-to-your-palma-site/" "palma-01" "demo/simple" "http://palma-01.your-institution.org"
```

Warning: This script was written in the context of upgrading our own machines and it might mess with your Debian package lists. Also it will set the default values for PIN (true), password (false) and policy link (none) in `palma.ini`. Please read and use said script with care. If in doubt, install PalMA manually as described below.

In the following we will cover the points you'll need to set up a PalMA station:

- Requirements
- Required Debian packages
- Webserver configuration
    - Apache2
    - Nginx-light
- PalMA
- Theming your installation
- Adding new languages

We assume that the web server's root directory is `/var/www/html` (default since Debian Jessie) and that PalMA will be installed directly there. Of course you can install PalMA in any other path.

All installation commands must be run as root user.

Requirements

For a PalMA station you need a computing device (e.g. a regular PC or a Raspberry Pi) with internet access and a monitor connected to it. The larger the screen, the greater the benefit.

PalMA runs on Linux (tested on Debian 9 Stretch and Raspbian), needs a webserver with PHP and SQLite and some viewer programs. Hardware requirements are relatively low. For reasonable performance we recommend something at least as strong as a Raspberry Pi 3.

Required packages

With the following lines we can install the needed viewer programs (for images, PDFs, videos and VNC connections), tools used for window management, database, PHP modules and building tools.

```
apt-get install midori feh vlc zathura ssvnc x11vnc
apt-get install wmctrl xdotool openbox libjs-jquery sqlite3
apt-get install php7.0 php7.0-cgi php7.0-cli php7.0-curl
apt-get install php7.0-fpm php7.0-gd php7.0-intl php7.0-sqlite3 php7.0-mbstring
apt-get install gettext git libavcodec-extra make
```

Now we install the webserver (normally apache2, but for Raspberry Pi we recommend `nginx-light`):

```
apt-get install apache2 libapache2-mod-php7.0
```

or

```
apt-get install nginx-light
```

Webserver configuration

Apache

The PHP default configuration for the Apache2 webserver permits file uploads up to 2 MB. This limit is too low for typical documents (images, office documents, pdf). Change the setting `upload_max_filesize in /etc/php/7.0/apache2/php.ini`. 10 MB is a good value. There is another limit for the maximum size of HTML posts with a default value of 8 MB. As this is less than the 10 MB needed for file uploads, the setting `post_max_size` must also be increased by setting it to 10 MB.

PalMA uses .htaccess to protect the database and the uploads directory. To enable this feature, Apache2 needs this section in file `/etc/apache2/sites-available/000-default.conf`:

```
<Directory /var/www/html>
    # "RewriteEngine" needs "FileInfo".
    # "Order" needs "Limit".
    AllowOverride FileInfo Limit
</Directory>
```

The Apache2 module rewrite must be enabled, too:

```
a2enmod rewrite
service apache2 restart
```

Nginx

For Raspberry Pi we replaced the apache2 web server with nginx because it uses much less resources. Make sure the following configurations (server root, enabling php7) are set in file **`/etc/nginx/sites-enabled/default`**:

```
server {
    root /var/www/html;
    index index.html index.htm index.php index.nginx-debian.html;
    # ...
    location ~ \.php$ {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
        }
}
```

PalMA

Now let's install what it's all about and get the latest version of PalMA from GitHub:

git clone **`https://github.com/UB-Mannheim/PalMA.git /var/www/html`**

Typically, PalMA should be started automatically. Activate **`autostart`** via **`systemd`** with these commands:

```
cp /var/www/html/scripts/palma.service /etc/systemd/system/palma.service
chmod 755 /etc/systemd/system/palma.service
systemctl daemon-reload
systemctl enable palma.service
```

Now a configuration file **`/var/www/html/palma.ini`** must be added. A template for this file is available from subdirectory examples, so run this command to get a preliminary file:

```
cp /var/www/html/examples/palma.ini /var/www/html/palma.ini
```

Please change entries in **`palma.ini`** according to your local installation. These include at least the entries **`theme`** and **`start_url`**.

Optionally we can create the language files for the translations of the user interface.

```
make -C /var/www/html
```

At last we need to grant write access to **`www-data`** so that the web server can create and modify the sqlite3 database **`palma.db`**, a directory for file uploads can be created automatically and some viewer programs can write their configuration data.

So we add write access for **`www-data`** in directory **`~www-data`** (typically **`/var/www`**) by changing the ownership:

```
chown -R www-data:www-data /var/www
```

Now you should have your own PalMA station up and running. See the next two sections on how to customize your installation and how to add new languages to it.

Theming your installation

PalMA was initially developed for the Learning Center at Mannheim University Library. So the looks of PalMA are coherent with our design. If you want to customize the design you can add a new theme. Add one or more directories for your institution in the directory theme, e.g.:

```
theme/our-institution/department1
theme/our-institution/department2
theme/some-other-institution
theme/your-institution
```

To change colors, icons and backgrounds in the user interface you will have to edit `palma.css` *and* `images/user_background.png`. *Better theming options might follow in future releases.*

Each theme directory must include these files:

- `background.png` - the background image on the team display. Any user windows will be shown on top of this background image. (Not to be confused with `images/user_background.png` that users see in the background of the user interface on their devices.)
- `favicon.ico` - icon typically shown in bookmark lists of browsers or when the PalMA URL is saved on a smartphone.
- `palma-logo-49x18.png` - logo used by PalMA's web interface `index.php`.
- `palma-logo-67x25.png` - logo used by the login web interface `login.php`.
- `screensaver.php` - is shown when no users are connected. It includes dynamically generated URL, PIN and QR-Code as well as background images. To prevent display burn-in it changes between two designs every few minutes. So we use different background images with English and German usage instructions:
    - `palma_d.png`
    - `palma_e.png`
- VNC software for screensharing
    - `winvnc-palma.exe` - an UltraVNC server for Windows, that **must be preconfigured to suit your institution**.
    - `VineServer.dmg` - a VNC software for Mac.
    - `x11.sh` - a script used for VNC screensharing on Linux.

Don't forget to enable your theme in `palma.ini`.

Add existing and new translations

PalMA initially supports English and German user interfaces for the web frontend. Please help us by providing additional translations for everyone on GitHub.

All translated texts are in the subdirectory **`locale`**. Newly added languages also need modifications in **`makefile`** and in **`i12n.php`**.

In a Debian GNU Linux installation, it is also necessary to add matching locales, either by running **`dpkg-reconfigure locales`** manually or by enabling the locales in /etc/locale.gen and running **`locale-gen`**. Here is an example which enables the English locale in its US variant (**`en_US.UTF-8`**):

```
perl -pi -e 's/^#.(en_US.UTF-8)/$1/' /etc/locale.gen && locale-gen
```