

---

# mpuls operations manual

Release 1.0

Intevation GmbH

December 11, 2013

## Contents

<b>1 Overview</b>	<b>1</b>
1.1 mpuls-Application . . . . .	1
1.2 Available Distributions . . . . .	2
1.3 Support . . . . .	2
<b>2 Basic Installation</b>	<b>2</b>
2.1 About the basic installation . . . . .	3
2.2 Requirements . . . . .	3
2.3 Setup mpuls . . . . .	3
2.4 Setup database . . . . .	4
2.5 Configure mpuls . . . . .	5
2.6 Running the application . . . . .	5

---

## 1 Overview

This documentation is part of the mpuls application which is released under the terms of the GNU Affero General Public License (AGPL) version 3 license.

### 1.1 mpuls-Application

An mpuls application consists of several different *modules*. There are modules for the web-application, some libraries, configuration files, documentation, evaluations, the database files, form definitions and scripts for administrating and maintaining the application.

There are two kinds of modules:

1. *Base modules*, shared by all mpuls applications
  - Administration scripts to setup databases
  - Maintenance scripts
  - Base webapplication

- Libraries.
- ...

## 2. *Application modules*, specific for each application

- Configuration
- Evaluations
- Datamodel
- Specifications
- ...

All those modules are managed with a Version Control System (VCS). Usually, all modules needed for an application are bundled together in some way, e.g. by a master repository referencing the module repositories or in a tarball for a release. These collections are called *distributions*.

## 1.2 Available Distributions

The following distributions are currently available

- mpuls WASKA
- mpuls WASKO
- mpuls WASKU
- mpuls WASKA 3
- mpuls mpuls-J

See the [download section](#) to download the distribution for installation.

## 1.3 Support

The [Intevation GmbH](#) offers support for a fee on

- Installation
- Operating (Service, Backups, Hosting, etc.)
- Development and Customizing
- User-Support (1st, 2nd and 3rd Level)
- Complex setups including evaluation servers and “AiR”<sup>1</sup> connectivity

Please send an email to [mpuls@intevation.de](mailto:mpuls@intevation.de) if you are interested in professional support or development.

## 2 Basic Installation

The steps below describe the installation process of a basic mpuls WASKA system on a *newly installed* Debian Squeeze system. The installations of other mpuls systems like mpuls WASKO or mpuls WASKU work in the same way, but some paths must be adapted.

Installation of mpuls on other Linux distributions than Debian Squeeze may work, but are *not tested*.

---

<sup>1</sup> The AiR feature enables different mpuls installation to share case data in a meta case.

## 2.1 About the basic installation

This installation guide will setup a basic mpuls WASKA application. This installation is not meant for real production use, but it is the base and starting point for your production setup. The next step towards a production setup is to host the application within a standard web server using the WSGI interface. We recommend using the Apache web server using the `mod_wsgi` module. An example wsgi script is included in the distribution. For this basic setup use a buildin webserver which ist meant for testing and development.

Building a production setup needs to address many aspects like backups, server and client SSL certificates, firewall configuration, web server configuration, logging etc. and depends on your environment and requirements. If you need assistance on setting up your application in a production environment you might be interested in our *Support* offers.

## 2.2 Requirements

Before installing the mpuls application we need to make sure that basic requirements are fulfilled.

On a freshly installed Debian Squeeze system you need to install the following packages:

```
apt-get install python-pylons \\  
python-pybabel python-openssl python-psycopg2 \\  
python-excelerator python-lxml \\  
gettext pwgen
```

If the server will also host the database we need the database packages as well and drop the automatically created cluster:

```
apt-get install postgresql-8.4 postgresql-plpython-8.4  
pg_dropcluster --stop 8.4 main
```

## 2.3 Setup mpuls

We will now setup the mpuls environment. mpuls will be installed in the home directory of a dedicated mpuls user. The following commands must be done as user “root”.

So first create a new user “mpuls”:

```
# as user "root"  
adduser --disabled-password --gecos "mpuls" mpuls
```

Downloaded the latest available version of your mpuls distribution (mpuls WASKA, mpuls WASKO, mpuls WASKU) from Wald ( [https://wald.intevation.org/frs/?group\\_id=40](https://wald.intevation.org/frs/?group_id=40) ) into the /tmp/ folder. The SSL certificate of Wald can be verified by using Intevation’s SSL CA:

```
cd /tmp  
wget https://ssl.intevation.de/Intevation-Root-CA-2010.crt  
wget --ca-certificate=Intevation-Root-CA-2010.crt \\  
https://wald.intevation.org/frs/download.php/1413/waska-2.1.2.tar.gz
```

**Note:** Files in /tmp will often be removed on reboot, therefore it would be good to keep a copy of the downloaded file elsewhere in case you need it later.

Now install the mpuls distribution with the following steps:

```

# Note that we use the name and version number from the downloaded
# file.
DISTRIBUTION=waska
REVISION=2.1.2
tar xzvf ${DISTRIBUTION}-${REVISION}.tar.gz
cd ${DISTRIBUTION}-${REVISION}
cd maintenance
sh install.sh -a -n ${DISTRIBUTION} -r ${REVISION}

# Continue with user "mpuls"
su - mpuls
cd /home/mpuls/waska/releases/current/base
python setup.py egg_info

```

## 2.4 Setup database

The setup of the database is described in detail in the documentation coming with the mpuls distribution sources in the “db-admin” folder. Here is the condensed version of the setup process.

As we want the new database to use a specific locale we need to reconfigure the locales of the setting and enable the needed locales using the following command. Do the following commands as user “root”:

```

dpkg-reconfigure locales

# No select the desired locales from the selection list. In this
# example we want a german UTF8 locale.
[*] de_DE.UTF-8 UTF-8

```

Now create a new database cluster with a specific locale setting where we will put the mpuls databases:

```

DISTRIBUTION=waska
pg_createcluster 8.4 ${DISTRIBUTION} --locale de_DE.UTF-8 --start

```

The command will report under which port the new created cluster will be reachable. Please save this port as we will need it in the next configuration steps. In this example the cluster is reachable on port 5432, which is the default port number for the first created cluster on the server. Now initialise the new cluster to be ready for mpuls databases:

```

# Continue with user postgres
su postgres
# Save the port in a environment variable
DBPORT=5432
cd /home/mpuls/waska/releases/current/db-admin/

```

Now initialise the new cluster to be ready for mpuls databases. The error message about a non-existing database can be ignored as it will be created during this step:

```

# As user "postgres"
python mpulsdbadm.py init_cluster \\\
-p ${DBPORT} \\\
--script-file=/home/mpuls/waska/releases/current/db-admin/sql/mpuls-init-cluster.sql \\\
--log-dir=/tmp

```

To create a new single database use the following command, you may change the name from “mydb” (use only a-z and 0-9) here and in the ini file below:

```
python mpulsdbadm.py create_db \  
-p ${DBPORT} \  
-d mydb \  
--script-dir=/home/mpuls/waska/releases/current/application-db/install \  
--log-dir=/tmp
```

Please note, that the command will create a “result\_list” file in the directory of the mpulsdbadm command. This list will include the initial passwords for the “adm” user in the database.

## 2.5 Configure mpuls

After we did a basic setup of the mpuls distribution and created a database, you need to create your own configuration file and optionally adjust settings (e.g. “mpuls.db.port” and “mpuls.db.name”) as user “mpuls”:

```
cd waska/releases/current/config/application  
cp demo.ini my.ini  
vi my.ini
```

## 2.6 Running the application

The basic setup is finished here. You should now be able to start the mpuls application. The following steps will enable you to test the mpuls installation using the “paster serve” command. This starts a minimalistic WSGI server using Python’s built-in BaseHTTPServer. The main purpose of this minimalistic server is testing and development of the mpuls application. Therefore we do not recommend the setup described for production use.

A production setup involves security aspects E.g. SSL-Encryption and other security measures, which depend on your individual security guidelines. Further we recommend to operate the server as WSGI applicaton in a robust webserver (E.g. Apache), which also depends on your local server specifications. Due to the broad variability of setups the installation of a production setup is beyond the scope of this installation guide. Please consult your computing support to setup a safe and secure web service.

However, to get the application up an running follow the next steps as user “mpuls”:

```
cd waska/releases/current/config/application  
BASEPATH=/home/mpuls/waska/releases/current/base/  
APPPATH=/home/mpuls/waska/releases/current/application/  
LIBFORMED=/home/mpuls/waska/releases/current/libformed/  
LIBMPULS=/home/mpuls/waska/releases/current/libmpuls/  
export PYTHONPATH=$BASEPATH:$APPPATH:$LIBFORMED:$LIBMPULS:$PYTHONPATH  
paster serve my.ini
```

The application should start now and shows an URL where it can be reached. Start your webbrower with this URL. You can log in to the application with the user “adm” and the generated password from the database setup.