



# **AFP2web Documentation**

## **AFP2web Server Version 1.x User's Guide and Reference**

Date: 2016-07-04

Maas Holding GmbH



**Copyright © 2016 Maas Holding GmbH, Germany**

All rights reserved. This publication and the concepts, solutions, programs described are copyright protected products of Maas Holding GmbH, Filderstadt, Federal Republic of Germany. This publication is subject to change or correction without prior notice.

Maas Holding GmbH will not be held liable for any damages caused or alleged to be caused either directly or indirectly by this publication.

No part of this documentation may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of Maas Holding GmbH.

## Table of Contents

<b>1</b>	<b>Preface to the AFPweb Server Guide .....</b>	<b>6</b>
<b>1.1</b>	<b>About AFP2web Server User's Guide and Reference .....</b>	<b>6</b>
1.1.1	Who Should Read this Document .....	6
1.1.2	What's in this Document .....	6
1.1.3	Where to Find Additional Information .....	7
1.1.4	Whom to Contact.....	7
<b>2</b>	<b>Introduction to the AFP2web Server - Overview .....</b>	<b>8</b>
<b>3</b>	<b>AFP2web Server User's Guide Reference .....</b>	<b>9</b>
<b>3.1</b>	<b>Prerequisites for the AFP2web Server.....</b>	<b>9</b>
3.1.1	System Prerequisites for the AFP2web Server .....	9
3.1.2	Software Prerequisites for the AFP2web Server .....	9
<b>3.2</b>	<b>AFP2web Server Installation.....</b>	<b>12</b>
<b>3.3</b>	<b>AFP2web Server Configuration .....</b>	<b>13</b>
3.3.1	Apache HTTP Server Configuration.....	13
3.3.2	AFP2web Ini File Configuration.....	18
3.3.3	Recommended Configuration Scenarios.....	18
<b>3.4</b>	<b>Log Files for the AFP2web Server .....</b>	<b>19</b>
3.4.1	error.log .....	19
3.4.2	access.log .....	19
<b>3.5</b>	<b>Testing the AFP2web Server .....</b>	<b>20</b>
3.5.1	Fetching the AFP2web Server's Version .....	20
3.5.2	Testing the AFP2web Server's Transform Service: .....	20
3.5.3	Testing the AFP2web Server's Get Status Service: .....	20
<b>3.6</b>	<b>AFP2web Server Integration .....</b>	<b>20</b>
3.6.1	Synchronous Transformation Service .....	21
3.6.2	Asynchronous Transformation Service.....	24
3.6.3	Get Status Service .....	27
3.6.4	Version Service .....	28
3.6.5	USID Service.....	29
<b>4</b>	<b>AFP2web Server Appendix .....</b>	<b>31</b>
<b>4.1</b>	<b>AFP2web Server Frequently Asked Questions.....</b>	<b>31</b>

4.1.1	Getting the connection reset error as given below when request is sent .....	31
4.1.2	Getting bad user name as given below while starting Apache HTTP server .....	31
4.1.3	Getting don't have permission to access /services/transform on this server as given below .....	31
4.1.4	How to define/extend Scripting Facility (or PERL engine) module search path?.....	31
<b>4.2</b>	<b>AFP2web Server Use Cases .....</b>	<b>32</b>
4.2.1	Case 1: ANY to PDF transformation: Signing output .....	32
4.2.2	Case 2: ANY to PDF transformation: Adding attachments to output .....	32
<b>List of Figures.....</b>		<b>34</b>
<b>List of Tables .....</b>		<b>35</b>
<b>Index</b>	<b>36</b>	

# 1 Preface to the AFPweb Server Guide

This preface offers

- a short introduction to the AFP2web Server
- information about who should read that document, where to find additional information and whom to contact for further issues
- information about the content of that document
- notices to be aware of before using the AFP2web Server

## 1.1 About AFP2web Server User's Guide and Reference

The AFP2web Server is an AFP2web-based solution. It a server program providing services meant to be used in a multiple requests environment. Since it is built on top of the well known AFP2web technology, it does provide the full functionality of AFP2web.

AFP2web is used for web-enabling, archiving, indexing, document exchange in workflows, for producing high-quality and true fidelity output, and as a component in a variety of application scenarios. The AFP2web Scripting Facility is an enhancement to AFP2web. It provides a scripting interface that is used to intelligently control document recognition, document splitting, index extraction, and much more.

### 1.1.1 Who Should Read this Document

This document is intended for all users of the AFP2web Server. To run the AFP2web Server, it is sufficient to set up an environment matching the hardware and software requirements and to configure the AFP2web Server. Users who need to customize AFP2web will find detailed information on how to do it within the AFP2web® Version 4.x User's Guide and Reference.

### 1.1.2 What's in this Document

The table below describes how this document is organized:

Chapter	Description
Introduction to the AFP2web Server	Gives an overview of the AFP2web server
Prerequisites	Describes the Prerequisites, the hardware requirements and the software to be installed before installing the AFP2web Server
AFP2web Server Installation	Describes the AFP2web Server installation procedure
Apache HTTP Server Configuration	Describes the configuration modifications required within the Apache HTTP Server
AFP2web Ini File	Describes the configuration modifications required within AFP2web
AFP2web Server Recommended Configuration	Describes the recommended AFP2web Server configuration for various scenarios

AFP2web Server Services	Describes the Services providing by the AFP2web Server for integrating it in your application
Testing the AFP2web Server	Give information about a testing the AFP2web Server installation

### 1.1.3 Where to Find Additional Information

To find detailed information about AFP2web's functionality, refer to the AFP2web® Version 4.x User's Guide and Reference. For further details about Apache HTTP Server, refer to <http://httpd.apache.org/>

### 1.1.4 Whom to Contact

This manual provides the information you need to get started with the AFP2web Server.

The Maas Holding GmbH offers professional services and consulting related to the development and deployment of AFP2web-based solutions. For more information, please visit our Web site at <http://www.oxseed.com/afp2web> or contact us via: E-mail: [afp2web@oxseed.com](mailto:afp2web@oxseed.com).

## 2 Introduction to the AFP2web Server – Overview

The AFP2web Server is a unified server meant to meet transformation requirements. Transformations are passed in through a service as an HTTP request to the AFP2web Server. The generated output will be, based on the service used for the request, either:

- sent back inline within the HTTP response
- or stored on a file system. In that case the status will be sent back within the HTTP response



## 3 AFP2web Server User's Guide Reference

### 3.1 Prerequisites for the AFP2web Server

List of prerequisites that are needed for the installation of the AFP2web Server

- System Prerequisites
- Software Prerequisites

For installation and configuration of Prerequisites and AFP2web Server, **admin privileges** are required.

#### 3.1.1 System Prerequisites for the AFP2web Server

The AFP2web Server is a 64-bit application that should run on any Linux systems matching the following requirements:

Description	Value
Operating System	Linux
Architecture	64 bit
Kernel Version	>= 2.6.18-6
Number of Cores	>= 4
Memory	>= 4 GB

#### 3.1.2 Software Prerequisites for the AFP2web Server

This chapter describes the list of the software that are to be installed before installing the AFP2web Server.

- Apache HTTP Server
- Apache module **mod\_fcgid**

##### 3.1.2.1 Apache HTTP Server Installation

Apache HTTP Server is a powerful, flexible, HTTP/1.1 compliant, highly configurable open-source web server, providing a secure and extensible platform that ensures HTTP service delivery in accordance with the current HTTP standards. AFP2web Server requires Apache HTTP Server version using the mpm-prefork engine. Following packages have to be installed in the order specified in the table below.

Package	Version	Comments
apache2-utils	>= 2.2.9-10	Utility programs for Apache HTTP Server

apache2.2-common	>= 2.2.9-10	Apache HTTP Server common files
apache2-mpm-prefork	>= 2.2.9	Apache HTTP Server - traditional non-threaded
Moreover the GLIBC version installed should be >= 2.3.6		

## Installation Example on Debian Linux 64 bit

To install Apache HTTP Server on Debian Linux 64 bit, follow the steps below.

- Ensure the system has internet access.
- Run the following commands in the specified order:

```
apt-get install apache2-utils
apt-get install apache2.2-common
apt-get install apache2-mpm-prefork
```

- If **apt-get** command fails due to missing packages on repositories, edit **/etc/apt/sources.list** file and add the following entries.

```
deb http://archive.debian.org/debian/ squeeze main contribnon-free
deb-src http://archive.debian.org/debian/ squeeze main contrib non-free
deb http://archive.debian.org/debian-security squeeze/updates main contribnon-free
deb-src http://archive.debian.org/debian-security squeeze/updates main contribnon-free
deb http://archive.debian.org/debian-volatile squeeze/volatile main contribnon-free
deb-src http://archive.debian.org/debian-volatile squeeze/volatile main contribnon-free
```

Use the equivalent appropriate commands to install Apache HTTP Server on other Linux variants.

If you install Apache Version 2.4 or above, then **apache2.x-common** might not be available and it can be ignored

### 3.1.2.2 Apache FastCGI Module

Apache FastCGI Module, **mod\_fcgid**, is a Apache plug-in module to high performance that starts a sufficient number instances of the FastCGI application to handle concurrent requests, and these applications remain running to handle further incoming requests. This module routes requests received by Apache HTTP Server to FastCGI application processes.

Following package has to be installed to get mod-fcgid on Linux System

Package	Version	Comments
---------	---------	----------

libapache2-mod-fcgid	>= 2.3.0	Apache 2 FastCGI module
----------------------	----------	-------------------------

### Installation Example on Debian Linux 64 bit

To install libapache2-mod-fcgid on Debian Linux 64 bit, follow the steps below.

- Ensure the system has internet access.
- Run the following command:

```
apt-get install libapache2-mod-fcgid
```

- Enable the installed fast cgi module

```
a2enmod fcgid
```

Use the equivalent appropriate commands to install Apache FastCGI Module on other Linux variants.

### 3.1.2.3 Running & Testing Apache HTTP Server

#### Running Apache HTTP Server

- To start Apache HTTP Server

```
/etc/init.d/apache2.start
```

- To stop Apache HTTP Server

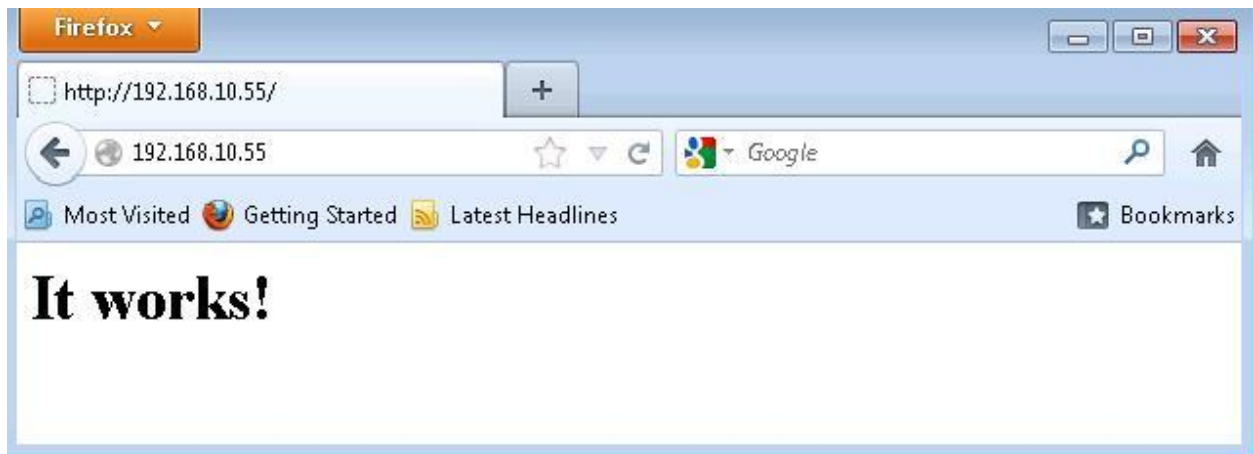
```
/etc/init.d/apache2 stop
```

- To restart Apache HTTP Server

```
/etc/init.d/apache2 restart
```

#### Testing Apache HTTP Server Installation

- Start the Apache HTTP Server
- Open a web browser on the installation server and type <http://localhost/> in the address bar or open a web browser on another machine and type <http://<IP Address of Apache Installed System>>
- You should get the text "It works!" message as shown below, for a successful Apache HTTP Server installation.



### 3.2 AFP2web Server Installation

This chapter describes the AFP2web Server release package as well as its installation.

- Change to the `lib/apache2/fcgid` directory.
- Create a `a2wserver` directory within `/var/lib/apache2/fcgid` directory.
- Set permission of the `a2wserver` directory.
- Change to the `lib/apache2/fcgid/a2wserver` directory.
- Extract the `afp2webServer_V1.x.x_linux_x64.tgz` package into the `/var/lib/apache2/fcgid/a2wserver` directory

```
cd /var/lib/apache2/fcgid/  
mkdir a2wserver  
chmod 777 a2wserver  
cd a2wserver  
tar xvzf <PKG_DOWNLOADED_PATH>/afp2webServer_V1.x.x_linux_x64.tgz
```

- Following files and directories will be extracted:

```
/var/lib/apache2/fcgid/a2wserver  
|-- /a2w  
|-- /afpcp  
|-- /conf  
|-- /doc  
|-- /extfont  
|-- /log  
|-- /pdf  
|-- /samples  
|-- /sfsamples  
|-- a2wtransformation.async  
|-- a2wtransformation.fcgi  
|-- afp2web.ini  
|-- afp2web.pm  
|-- history.txt  
|-- index.html  
|-- getstatus.html  
|-- liba2wsdkXX.so  
|-- libcrypto.so.X.0
```

```
-- libcurl.so.X
-- libperl.so.X.X
-- libssl.so.X.X
-- libstdc++.so.X
-- libz.so.X
-- license_en.txt
-- readme.txt
-- server.log.prop
-- transform.html
-- version.html
```

- Run the following commands to set user, user group for the extracted files and to assign needed permissions for a2wtransformation.async and a2wtransformation.fcgi file.

```
cd /var/lib/apache2
chown -R www-data:www-data *
cd /var/lib/apache2/fcgid/a2wserver
chmod 777 a2wtransformation.fcgi
chmod 777 a2wtransformation.async
```

- Run the following command to create a link that would make the Perl library accessible to the AFP2web Server.

```
ln -s /var/lib/apache2/fcgid/a2wserver/libperl1.so.5.8 /usr/lib/libperl1.so.5.8
```

### 3.3 AFP2web Server Configuration

#### 3.3.1 Apache HTTP Server Configuration

To do the following configuration steps admin privileges are required.

Apache Server can be configured either manually or by copying configuration files in release package to specific Apache directories.

1. Refer following sections to manually configure Apache HTTP Server.
  - Set up FastCGI module and Aliases
  - Configuring the Apache Prefork Module
  - FCGI Applications Configuration
2. Follow the steps below to copy configuration files in release package to specific Apache directories.
  - AFP2web Server Release package `afp2webServer_V1.x.x.x_linux_x64.tgz` contains configurations files in "conf" sub directory. Apache HTTP Server can be configured by copying these files into specific Apache directories as described below.
  - Copy `conf/apache2.conf` file in `afp2webServer_V1.x.x.x_linux_x64.tgz` to `/etc/apache2/` directory.
    - ```
cd /var/lib/apache2/fcgid/a2wserver/conf
cp apache2.conf /etc/apache2
```
  - Copy `conf/fcgid.conf` file in `afp2webServer_V1.x.x.x_linux_x64.tgz` to `/etc/apache2/mods-available` directory.
    - ```
cd /var/lib/apache2/fcgid/a2wserver/conf
cp fcgid.conf /etc/apache2/mods-available
```

- Copy conf/default file in afp2webServer\_V1.x.x.x\_linux\_x64.tgz to /etc/apache2/sites-available file directory
  - `cd /var/lib/apache2/fcgid/a2wserver/conf`  
`cp default /etc/apache2/sites-available`

### 3.3.1.1 Set up FastCGI module and Aliases

In order to set up FastCGI module and aliases, entries marked by the "AFP2web Server Begin" and "AFP2web Server End" tags are added in `/etc/apache2/sites-available/default` file as shown below:

Code Block 1 `/etc/apache2/sites-available/default`

```
<VirtualHost *:8080>
....
....
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>

#---- AFP2web Server Begin ----#
#---- Alias for version
Alias /services/version
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi

#---- Alias for unique system id
Alias /services/usid
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi

#---- Alias for synchronous transformation
Alias /services/transform
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi

#---- Alias asynchronous transformation and get status
Alias /services/asyncTransform
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.async
Alias /services/getStatus
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi

#---- Alias for root of services
Alias /services /var/lib/apache2/fcgid/a2wserver

ScriptAlias /fcgi-bin/ /var/lib/apache2/fcgid/a2wserver/
<Directory "/var/lib/apache2/fcgid/a2wserver/">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
#---- AFP2web Server End ----#

ErrorLog /var/log/apache2/error.log
</VirtualHost>
```

### 3.3.1.2 Configuring the Apache Prefork Module

Apache uses following directives in the `/etc/apache2/apache2.conf` file to configure the prefork module. The parameters of that directive specify how the forked servers have to be handled.

Code Block 2 `/etc/apache2/apache2.conf`

```
<IfModule mpm_prefork_module>
    StartServers      4
    MinSpareServers   4
    MaxSpareServers   10
    MaxClients        256
    MaxRequestsPerChild 0
</IfModule>
```

The table below described the parameters of this directive.

Parameter	Description
StartServers	The StartServers parameter defines the number of forked servers created at startup.
MinSpareServers	The MinSpareServers parameter defines the minimum number of forked servers that still should run when no load exists at all. An idle forked server is one which is not handling a request.
MaxSpareServers	The MaxSpareServers parameter defines the maximum number of forked servers that still should run when no load exists at all. If more than MaxSpareServers servers are idle, then the main server will stop the exceeding servers.
MaxClients	The MaxClients parameter defines the maximum of simultaneous requests that will be served.
MaxConnectionsPerChild	The MaxConnectionsPerChild parameter defines the maximum of connections that a forked server can handle. Once MaxConnectionsPerChild connections are reached, the forked server will expire. If MaxConnectionsPerChild is 0, then the server will never expire.
How to properly set these parameters depends on the expected load. Refer to the "Recommended Configurations Scenarios" chapter for further details.	

### 3.3.1.3 Port Configuration

By default, Apache Server listens on port 80 for incoming requests. Sometimes it is possible that this port has already been taken by other application. In such a case, Apache Server needs to be configured to listen on different port for example 8080. To do so, follow the below steps OR copy `/var/lib/apache2/fcgid/a2wserver/conf/ports.conf` to `/etc/apache2`.

- Modify port number in `/etc/apache2/ports.conf`

#### Code Block 3 /etc/apache2/ports.conf

```
NameVirtualHost *:8080
Listen 8080
```

- Modify port number in /etc/apache2/sites-available/default

#### Code Block 4 /etc/apache2/sites-available/default

```
<VirtualHost *:8080>
....
....
</VirtualHost>
```

### 3.3.1.4 FCGI Applications Configuration

FCGI applications start on demand. Additional instances will be started automatically if load increases. Following FastCGIConfig directive in **/etc/apache2/mods-available/fcgid.conf** is used to control this process.

#### Code Block 5 /etc/apache2/mods-available/fcgid.conf

```
<IfModule mod_fcgid.c>
  AddHandler fcgid-script .fcgi
  DefaultMinClassProcessCount 128
  DefaultMaxClassProcessCount 256
  MaxRequestsPerProcess 256
  MaxRequestLen 5242880
  IdleTimeout 600
  BusyTimeout 300
  ErrorScanInterval 60
  DefaultInitEnv LD_LIBRARY_PATH=./var/lib/apache2/fcgid/a2wserver
  DefaultInitEnv PERLLIB=./usr/local/perl/lib
</IfModule>

<IfModule mod_fcgid.c>
  AddHandler fcgid-script .async
  DefaultMinClassProcessCount 128
  DefaultMaxClassProcessCount 256
  MaxRequestsPerProcess 256
  IdleTimeout 600
  BusyTimeout 600
  ErrorScanInterval 60
  DefaultInitEnv LD_LIBRARY_PATH=./var/lib/apache2/fcgid/a2wserver
  DefaultInitEnv PERLLIB=./usr/local/perl/lib
</IfModule>
```

The table below described the parameters of this directive.

Parameter	Description
-----------	-------------



DefaultMinClassProcessCount	Minimum process count of one class of FastCGI application. This directive sets the minimum number of processes that will be retained in a process class after finishing requests.												
DefaultMaxClassProcessCount	Maximum process count of one class of FastCGI application. This directive sets the maximum number of processes that can be started for each process class.												
MaxRequestsPerProcess	<div>Maximum requests handled by each FastCGI application. FastCGI application processes will be terminated after handling the specified number of requests. A value of 0 disables the check.</div> <div>A value of -1 is currently accepted for ease of migration for existing configurations. It is treated the same as 0.</div>												
MaxRequestLen	<div>Maximum request length in bytes. If the request body exceeds this size, the request will fail with 500 Server Error.</div> <div>If this parameter is not specified, server uses the default value of 128KB. AFP2web server recommends the value of 5MB (5242880 bytes) as given above, still value can be changed by administrators based on their application requirements.</div>												
IdleTimeout	An idle FastCGI application will be killed after IdleTimeout. Application processes which have not handled a request for this period of time will be terminated, if the number of processes for the class exceeds DefaultMinClassProcessCount. A value of 0 disables the check.												
BusyTimeout	A FastCGI application will be killed after handling a request for BusyTimeout. This is the maximum time limit for request handling. If a FastCGI request does not complete within BusyTimeout seconds, it will be subject to termination.												
ErrorScanInterval	Scan interval for exited process. This is the interval at which the module will handle pending process termination. Termination is pending for any processes which have exceeded IdleTimeout or ProcessLifeTime												
DefaultInitEnv	<div>A name=value pair to define environment variables to pass to the FastCGI application. This directive can be used multiple times.</div> <div>Some common environment variables to be defined for AFP2web Server are listed below</div> <table><thead><tr><th>Variable Name</th><th>Value</th><th>Mandatory</th><th>Description</th></tr></thead><tbody><tr><td>LD_LIBRARY_PATH</td><td>List of paths separated by ':'</td><td>Yes</td><td>Specifies the search path for loading dynamic libraries. This path must include the current directory(".")</td></tr><tr><td>PERLLIB</td><td>List of paths separated by ':'</td><td>No</td><td>Specifies the search path for loading PERL modules. Additional PERL module search paths for Scripting Facility can be configured here.</td></tr></tbody></table>	Variable Name	Value	Mandatory	Description	LD_LIBRARY_PATH	List of paths separated by ':'	Yes	Specifies the search path for loading dynamic libraries. This path must include the current directory(".")	PERLLIB	List of paths separated by ':'	No	Specifies the search path for loading PERL modules. Additional PERL module search paths for Scripting Facility can be configured here.
Variable Name	Value	Mandatory	Description										
LD_LIBRARY_PATH	List of paths separated by ':'	Yes	Specifies the search path for loading dynamic libraries. This path must include the current directory(".")										
PERLLIB	List of paths separated by ':'	No	Specifies the search path for loading PERL modules. Additional PERL module search paths for Scripting Facility can be configured here.										

How to properly set these parameters depends on the expected load. Refer to the "Recommended Configurations Scenarios" chapter for further details.

Refer "[https://httpd.apache.org/mod\\_fcgid/mod/mod\\_fcgid.html](https://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html)" webpage for more details about this configuration options. Above config option names are valid up to mod\_fcgid version <=2.3.0. For mod\_fcgid version >2.3.0, please use the new option names found on this webpage explained under "[Update Notes](#)" topic.

### 3.3.2 AFP2web Ini File Configuration

This configuration file "afp2web.ini" defines global parameters of AFP2web, such as conversion options, output paths, paths to AFP resources, and other processing options. For a complete description, please refer to Section INI File Parameters (afp2web.ini) of the AFP2web V4.x User's Guide and Reference.

Ensure to set the "Licensee" and "SerialNr" parameters as determined by Maas Holding GmbH as shown below:

```
;-----  
; The name of licensee as determined by Maas Holding GmbH  
; Set the value within quotes  
;-----  
Licensee="My Company LIN|input|Output Options"  
;-----  
; The serial number as determined by Maas Holding GmbH  
; Set the value within quotes  
;-----  
SerialNr="3C609Z94-4AZD22B9"
```

Both "Licensee" and "SerialNr" will activate the transformations and the platform, which have been purchased.

### 3.3.3 Recommended Configuration Scenarios

Based on results of some benchmark tests conducted by the AFP2web team, we recommend the following configuration scenarios based on the load you expect. The goal of these scenarios is to deliver an average response of less than a second for synchronous transformation of a simple 3 pages AFP document to PDF document.

The following table describes the recommended configuration scenarios for 10, 50 and 100 users

Max. Users - Max. Requests	Apache Configuration (/etc/apache2/ apache2.conf)	FCGI Configuration (/etc/apache2/ mods-enabled/ fcgid.conf)	System Configuration
100 Users –	StartServers: 256 MinSpareServers: 256 MaxSpareServers: 256	DefaultMinClassProcessCount: 128 DefaultMaxClassProcessCount: 256 MaxRequestsPerProcess: 256	OS: Debian 5.0, 64 bit Linux Number of Cores: <b>8</b>

200 Requests	MaxClients: 256 MaxRequestsPerChild: 0	IdleTimeout: 600 BusyTimeout: 300 ErrorScanInterval: 60	Memory: <b>8 GB</b>
50 users – 200 requests	StartServers: 256 MinSpareServers: 256 MaxSpareServers: 256 MaxClients: 256 MaxRequestsPerChild: 0	DefaultMinClassProcessCount: 128 DefaultMaxClassProcessCount: 256 MaxRequestsPerProcess: 256 IdleTimeout: 600 BusyTimeout: 300 ErrorScanInterval: 60	OS: Debian 5.0, 64 bit Linux Number of Cores: <b>8</b> Memory: <b>8 GB</b>
10 users – 200 requests	StartServers: 256 MinSpareServers: 256 MaxSpareServers: 256 MaxClients: 256 MaxRequestsPerChild: 0	DefaultMinClassProcessCount: 100 DefaultMaxClassProcessCount: 200 MaxRequestsPerProcess: 200 IdleTimeout: 600 BusyTimeout: 300 ErrorScanInterval: 60	OS: Debian 5.0, 64 bit Linux Number of Cores: <b>4</b> Memory: <b>4 GB</b>
Refer " <a href="https://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html">https://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html</a> " webpage for more details about "mod_fcgid" configuration options			

Ensure to restart the Apache HTTP Server, after configuration changes are made

### 3.4 Log Files for the AFP2web Server

There are two log files in /var/log/apache2/ directory.

#### 3.4.1 error.log

- AFP2web Server logs information about request processing and also the errors occurred during processing the request in this file.
- In case of error, this file needs to be analyzed to find the cause of error.

#### 3.4.2 access.log

- This log file records all requests processed by the Apache HTTP server.
- This file can be analyzed to get useful statistics like number of request/second

### 3.5 Testing the AFP2web Server

The AFP2web Server release package comes with 2 HTML forms to help you test the AFP2web Server.

- The Version Service form
- The Transform Service form

Before start testing, ensure Apache Server is started and running. To start Apache Server, refer chapter Running & Testing Apache HTTP Server in section 3.1.

#### 3.5.1 Fetching the AFP2web Server's Version

This form is meant for fetching information about the AFP2web Server. Hence ensuring that the AFP2web Server has been successfully

- Type in your favorite browser `http://<server>:<port>/services/version.html`
- Set the options you need
- Click the "GetVersion" button

#### 3.5.2 Testing the AFP2web Server's Transform Service:

This form is meant for testing the AFP2web Server's Transform Service.

- Type in your favorite browser `http://<server>:<port>/services/transform.html`
- Select a file and modify the parameters as you wish
- Click the "Transform" button to do synchronous transformation
- Click the "Asynchronous Transform" button to do asynchronous transformation

#### 3.5.3 Testing the AFP2web Server's Get Status Service:

This form is meant for getting status of the asynchronous transformation.

- Type in your favorite browser `http://<server>:<port>/services/getstatus.html`
- Type in the JobId of the asynchronous transformation request.
- Click the "Get Job Status" button

### 3.6 AFP2web Server Integration

The AFP2web Server provides Services to facilitate the integration of its functionality in your applications. The Services use the HTTP protocol for request and responses. These Services are:

- Synchronous Transformation Service: is meant to transform documents synchronously. Useful for on the fly conversions.
- Asynchronous Transformation Service: is meant to transform documents asynchronously. Useful for spool conversions that take long time for conversion.
- Get Status Service: is meant to get status of asynchronous transformation.
- Version Service: use it to retrieve information about the AFP2web Server and its component

This chapter describes these Services.

You can also get a quick usage reference of these Services by typing in your favorite browser `http://<server>:<port>/services`.

### 3.6.1 Synchronous Transformation Service

This Transformation API is used for transforming the documents to various output formats synchronously. Useful for on the fly conversions.

Input document can be any one of the following.

- Buffer
- File name
- List of file names
- URL
- List of URLs

Output document can be sent to any one of the following destinations.

- As HTML with output document buffer. Default.
- As files on File System

#### 3.6.1.1 URL

*POST method: `http://<server ip>[:<service port>]/services/transform`*

#### Request Parameters

Component	Parameter	Value	Description
server	server_log_level	OFF, INFO (default), WARN, ERROR, DEBUG.	Set server log level
	ResponseFormat	HTML   JSON.	Requested response format. <div>JSON is NOT supported when output is returned as Buffer</div>

	InputBuffer	<buffer>	Content of file to be transformed
	OR		
	InputURLs	Comma separated list of absolute input file names or URLs	<p>One or more input file name(s) or URL(s) to be transformed</p> <div>When file name is given, the file must be accessible by AFP2web Server.</div> <div>Only <b>http://</b> URL protocol is supported</div>
	OutputMode	File   Buffer (default)	<p><b>Buffer</b> means transformed output document sent back as binary data with proper mime type on response</p> <p><b>File</b> means transformed output document(s) will be written to output file path (specified by parameter "<b>OutputFilePath</b>"). This path must be accessible/writable from AFP2web Server.</p>
	OutputFormat	Output file format	Expected output file format. Optional parameter and Default is PDF.
	OutputFilePath	Absolute path for output	Expected path where transformed output document(s) should be written. This path

			must be writable from AFP2web Server
<b>AFP2web</b>  Parameters of AFP2web must be passed in with following syntax: <b>a2w_&lt;section name&gt;_&lt;parameter&gt;=&lt;value&gt;</b> where - Possible section is <b>settings</b> - Possible settings parameters are described in <a href="#">AFP2web User Guide</a> under chapter "AFP2web Reference->INI Parameter Reference->Section [Settings]"	a2w_settings_QuietMode	On   Off (default)	Turn on/off console messages
	a2w_settings_Color	On   Off (default)	Turn on/off color output
<b>HTML2X</b>  Parameters of HTML2X must be passed in with following syntax: <b>h2x_&lt;section name&gt;_&lt;parameter&gt;=&lt;value&gt;</b> where - Possible sections are <b>global</b> and <b>object</b> - Possible global parameters are listed <a href="#">here</a> - Possible object parameters are listed <a href="#">here</a>	h2x_global_colorMode	Color   Grayscale (default)	Generate color or gray scale output
	h2x_object_web.enableIntelligentShrinking	true (default)   false	Enable intelligent shrinking to fit more content on one page

## Response

### Response when OutputMode is "Buffer" (Default case).

- Transformed output document will be sent as HTML with proper mime type.

### Response when OutputMode is "File"

- Output document(s) are stored on path specified in "**OutputFilePath**" parameter.
- After process is over, following response will be sent to the caller either as HTML or as JSON

PID	<Process Id>
RequestNumber	<Request Id>

ApplicationName	AFP2web Server
Response Time	<Response time in milliseconds>
Status	Success
ReturnCode	0

### Response in case of error

- In case of error, error response with error code and message will be sent to caller as HTML or as JSON

PID	<Process Id>
RequestNumber	<Request Id>
ApplicationName	AFP2web Server
Response Time	<Response time in milliseconds>
Status	Error, <Error Message>
ReturnCode	<Error Code>

## 3.6.2 Asynchronous Transformation Service

This Transformation API is used for transforming the documents to various output formats asynchronously. Useful for pool conversions that take long time for conversions.

This service will,

- Accept input spool file name and a unique Job Id for the asynchronous request
- Converts input spool file
- Stores transformed output documents on File System on given output file path.

### 3.6.2.1 URL

**POST method:** *http://<server ip>[:<service port>]/services/asyncTransform*

### Request Parameters

Component	Parameter	Value	Description
server	server_log_level	OFF, INFO (default),	Set server log level



		WARN, ERROR, DEBUG.	
	ResponseFormat	HTML   JSON.	Requested response format.
	JobId	Unique job identifier string	Unique job identifier used to distinguish multiple asynchronous transformation calls and to get status of particular asynchronous transformation later.
	InputURLs	Comma separated list of absolute input file names or URLs	<p>One or more input file name(s) or URL(s) to be transformed</p> <div>When file name is given, the file must be accessible by AFP2web Server.</div> <div>Only <b>http://</b> URL protocol is supported</div>
	OutputMode	File (default)	<b>File</b> means transformed output document(s) will be written to output file path (specified by parameter " <b>OutputFilePath</b> "). This path must be accessible/writable from AFP2web Server.
	OutputFormat	Output file format	Expected output file format. Optional parameter and Default is PDF.
	OutputFilePath	Absolute path for output	Expected path where transformed output document(s) should be written. This path must be writable from AFP2web Server

	StatusCallback	POST URL which will be invoked with transformation status after completion.	Optional, POST parameters passed in to callback are:	
			<b>Parameter</b>	<b>Description</b>
			JobId	Unique Job Identifier
			JobPID	Process id of Server application processing transformation request
			Filename	List of input files passed in to asynchronous request
<b>AFP2web</b> Parameters of AFP2web must be passed in with following syntax: <b>a2w_&lt;section name&gt;_&lt;parameter&gt;=&lt;value&gt;</b> where - Possible section is <b>settings</b> - Possible settings	a2w_settings_QuietMode	On   Off (default)	Turn on/off console messages	
	a2w_settings_Color	On   Off (default)	Turn on/off color output	

parameters are described in <a href="#">AFP2web User Guide</a> under chapter "AFP2web Reference->INI Parameter Reference->Section [Settings]"			
<b>HTML2X</b> Parameters of HTML2X must be passed in with following syntax: <b>h2x_&lt;section name&gt;_&lt;parameter&gt;=&lt;value&gt;</b> where - Possible sections are <b>global</b> and <b>object</b> - Possible global parameters are listed <a href="#">here</a> - Possible object parameters are listed <a href="#">here</a>	h2x_global_colorMode	Color   Grayscale (default)	Generate color or gray scale output
	h2x_object_web.enableIntelligentShrinking	true (default)   false	Enable intelligent shrinking to fit more content on one page

## Response

- Asynchronous transformation request will return a HTML response with empty body content after the conversion is over. The client need not wait for this response.
- To get status of the asynchronous transformation, client must send [Get Status](#) request with **JobId** of asynchronous transformation **OR** use '**StatusCallback**' to get the status of asynchronous transformation.

### 3.6.3 Get Status Service

This API gives information about the status of the asynchronous transformation.

#### 3.6.3.1 URL

*GET method: [http://<server ip>\[:<service port>\]/services/getStatus?JobId=<JobId>](http://<server ip>[:<service port>]/services/getStatus?JobId=<JobId>)*

## Request Parameters

Component	Parameter	Value	Description
server	server_log_level	OFF, INFO (default), WARN, ERROR, DEBUG.	Set server log level
	ResponseFormat	HTML   JSON.	Requested response format.

	JobId	Unique job identifier string	Unique job identifier to get status of particular asynchronous transformation.
--	-------	------------------------------	--

## Response

- Response in case of success

PID	<Process Id>
RequestNumber	<Request Id>
ApplicationName	AFP2web Server
JobID	<Job ID>
JobPID	<Process Id of asynchronous transformation job>
Status	"Working" "Done"
ReturnCode	0

- Response in case of error

PID	<Process Id>
RequestNumber	<Request Id>
ApplicationName	AFP2web Server
JobID	<Job ID>
JobPID	<Process Id of asynchronous transformation job>
Status	Error, <Error Message>
ReturnCode	<Error Code>

### 3.6.4 Version Service

This API gives information about version information about AFP2web Server and its components

#### 3.6.4.1 URL

*POST method: [http://<server ip>\[:<service port>\]/services/version](http://<server ip>[:<service port>]/services/version)*

## Request Parameters

Component	Parameter	Value	Description
server	server_log_level	OFF, INFO (default), WARN, ERROR, DEBUG.	Set server log level
	ResponseFormat	HTML   JSON.	Requested response format.
	version	On (default)   Off	Version of server
	version_all	On   Off (default)	Version of server and all it's modules
	a2w	On   Off (default)	Version of AFP2web module
	a2w_all	On   Off (default)	Version of AFP2web module and all it's components
	h2x	On   Off (default)	Version of htmltox module
	h2x_all	On   Off (default)	Version of htmltox module and all it's components

## Response

- Response in case of success

Version information returned as response

- Response in case of error

PID	<Process Id>
RequestNumber	<Request Id>
ApplicationName	AFP2web Server
Status	<Error Message>
Code	<Error Code>

### 3.6.5 USID Service

This API gives the Unique System ID of AFP2web server for licensing purpose.

#### 3.6.5.1 URL

*GET method: `http://<server ip>[:<service port>]/services/usid`*

## Request Parameters

Component	Parameter	Value	Description
server	server_log_level	OFF, INFO (default), WARN, ERROR, DEBUG.	Set server log level
	ResponseFormat	HTML   JSON.	Requested response format.

## Response

- Response in case of success contains Unique System ID (USID)
- Response in case of error

PID	<Process Id>
RequestNumber	<Request Id>
ApplicationName	AFP2web Server
Status	Error, <Error Message>
ReturnCode	<Error Code>

## 4 AFP2web Server Appendix

### 4.1 AFP2web Server Frequently Asked Questions

#### 4.1.1 Getting the connection reset error as given below when request is sent

Connection to the server was reset while the page was loading.

This occurs if Apache HTTP server is not running. Start the Apache Server with the following command/`etc/init.d/apache2 start`

#### 4.1.2 Getting bad user name as given below while starting Apache HTTP server

apache2: bad user name \${APACHE\_RUN\_USER}

This occurs if User and Group are not set in `/etc/apache2/apache2.conf` file.

Search for the following line in `/etc/apache2/apache2.conf` file and comment them **#User**  
**\${APACHE\_RUN\_USER} #Group \${APACHE\_RUN\_GROUP}** Instead add following lines below them **User www-data**  
**Group www-data**

#### 4.1.3 Getting don't have permission to access /services/transform on this server as given below

*Forbidden You don't have permission to access /services/transform on this server. Apache/2.2.9 (Debian) mod\_fcgid Server at 192.168.10.55 Port 80*

This error occurs if `/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi` does not have executable permission. Run the following command to set the permission

`chmod 777 a2wtransformation.fcgi`

#### 4.1.4 How to define/extend Scripting Facility (or PERL engine) module search path?

AFP2web Server by default searches Scripting Facility modules in AFP2web Server's installation directory. In order to add your own Scripting Facility modules paths, you need to extend the list of paths defined in the **PERLLIB** environment variable of the FastCGI Configuration (`/etc/apache2/mods-available/fcgid.conf`), as shown below

`/etc/apache2/mods-available/fcgid.conf`

```
<IfModule mod_fcgid.c>
  AddHandler fcgid-script .fcgi
  ...
  DefaultInitEnv LD_LIBRARY_PATH=:/var/lib/apache2/fcgid/a2wserver
  DefaultInitEnv PERLLIB=:/usr/local/perl/lib
</IfModule>

<IfModule mod_fcgid.c>
  AddHandler fcgid-script .async
  ...
  DefaultInitEnv LD_LIBRARY_PATH=:/var/lib/apache2/fcgid/a2wserver
  DefaultInitEnv PERLLIB=:/usr/local/perl/lib
</IfModule>
```

*NOTE: Within the **PERLLIB** environment variable, separator is ':'.*

## 4.2 AFP2web Server Use Cases

### 4.2.1 Case 1: ANY to PDF transformation: Signing output

This section describes all the parameters needed to sign PDF output

Parameter	Value	Mandatory	Description
OutputFormat	PDF	Yes	PDF output format selected to sign output
Sign	On	Yes	Turn on signing PDF output
SignCertificateFile	Certificate File	Yes	PKCS12 Certificate File name issued by a Certificate Authority to sign output.
SignPassword	Encrypted password	Yes	Encrypted password, used to decrypt the <b>PKCS #12</b> Certificate file. Refer AFP2web User Guide documentation about this parameter to know how to encrypt plain password.
SignName	Name	No	Name of the person or authority signing the output document
SignLocation	Location	No	Location from where the output document is signed
SignReason	Reason	No	Reason for signing the output document

### 4.2.2 Case 2: ANY to PDF transformation: Adding attachments to output

This section describes all the parameters needed to add attachments on PDF output

Parameter	Value	Mandatory	Description
OutputFormat	PDF	Yes	PDF output format selected to add attachments
Attachments	[{"filename:<filename>","desc:<description>"}]	Yes	List of the files to be attached to the PDF output, must be specified as JSON array.  <b>Example:</b> Attachments=[{"filename":"/home/user/attachments/a1.txt","desc":"Text File"}, {"filename":"att/a2.zip","desc":"Zip file"}]
PageMode	UseAttachments	No	An option specifying how the document should be displayed when opened.



			<b>UseAttachments</b> , Open the document and display the attachments panel
--	--	--	---

## List of Figures

No table of figures entries found.

## List of Tables

No table of figures entries found.

## **I n d e x**

No index entries found.