



AFP2web Server Version 1.x

User's Guide and Reference

Version 1.x

2015-Jul-23 14:30

AFP2web Team

Table of Contents

Preface to the AFPweb Server Guide	3
About AFP2web Server User's Guide and Reference	3
Introduction to the AFP2web Server - Overview	4
AFP2web Server User's Guide Reference	4
Prerequisites for the AFP2web Server	4
System Prerequisites for the AFP2web Server	4
Software Prerequisites for the AFP2web Server	4
Apache HTTP Server Installation	4
Apache FastCGI Module	5
Running & Testing Apache HTTP Server	5
AFP2web Server Installation	6
AFP2web Server Configuration	7
Apache HTTP Server Configuration	7
AFP2web Ini File Configuration	11
Recommended Configuration Scenarios	11
Log Files for the AFP2web Server	12
Testing the AFP2web Server	12
AFP2web Server Integration	13
Synchronous Transformation Service	13
Asynchronous Transformation Service	14
Get Status Service	15
Version Service	16
USID Service	17
AFP2web Server Additional Information	17
AFP2web Server Frequently Asked Questions	17

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

About AFP2web Server User's Guide and Reference

The AFP2web Server is an AFP2web-based solution. It is 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.

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.

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

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/>

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

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

information, please visit our Web site at <http://www.oxseed.com/afp2web> or contact us via: E-mail: afp2web@oxseed.com.

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

AFP2web Server User's Guide Reference

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.

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

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

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

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

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.

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.

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

Running & Testing Apache HTTP Server

Running Apache HTTP Server

- To start Apache HTTP Server

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

- To stop Apache HTTP Server

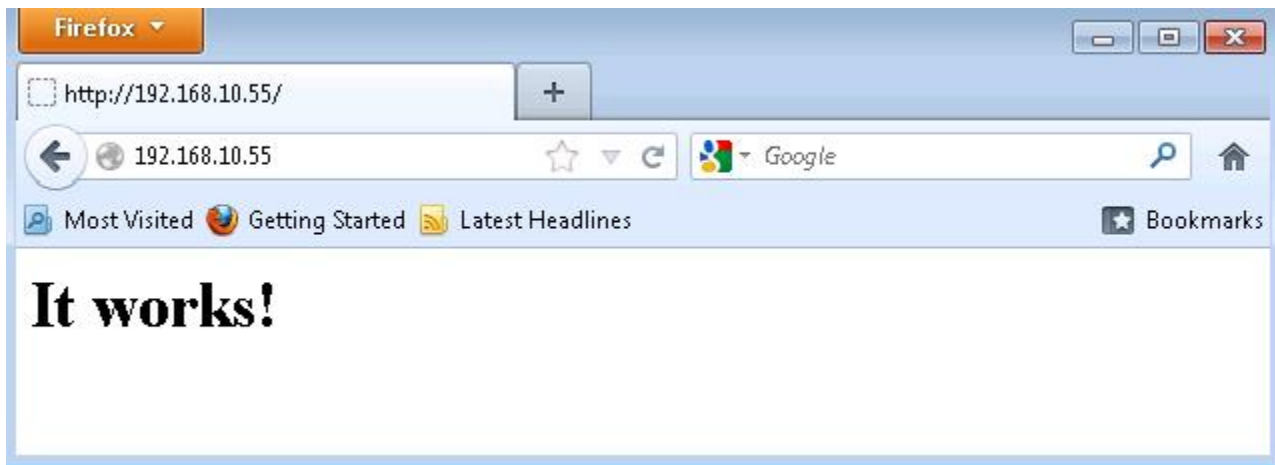
```
| /etc/init.d/apache2 stop
```

- 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.



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.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.x_linux_x64.tgz
```

- Following files and directories will be extracted:

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

```
/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
|-- liba2wsdk43.so
|-- libcurl.so.4
|-- libperl.so.5.8
|-- libstdc++.so.5
|-- license_en.txt
|-- readme.txt
|-- 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/libperl.so.5.8 /usr/lib/libperl.so.5.8
```

AFP2web Server Configuration

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
- CGI 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`

Set up FastCGI module and Aliases

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

Add the entries marked by the "AFP2web Server Begin" and "AFP2web Server End" tags into the `/etc/apache2/sites-available/default` file as shown below:

```
/etc/apache2/sites-available/default

<VirtualHost *:80>
    ....
    ....
    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 synchronous transformation
    Alias /services/transform
/var/lib/apache2/fcgid/a2wserver/a2wtransformation.fcgi

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

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

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

    ScriptAlias /cgi-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>
```

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.


```
/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.

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.

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

```
/etc/apache2/ports.conf
```

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

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

```
/etc/apache2/sites-available/default
```

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

FCGI Applications Configuration

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

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.

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

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

`LD_LIBRARY_PATH` given above specifies the search path for loading dynamic libraries. This path must include the current directory(".").

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	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>A value of -1 is currently accepted for ease of migration for existing configurations. It is treated the same as 0.</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	A name=value pair to define environment variables to pass to the FastCGI application. This directive can be used multiple times.

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

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" 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.

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.

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 - 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: 8 Memory: 8 GB
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: 8 Memory: 8 GB

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

10 users	StartServers: 256	DefaultMinClassProcessCount: 100	OS: Debian 5.0, 64 bit Linux
–	MinSpareServers: 256	DefaultMaxClassProcessCount: 200	Number of Cores: 4
200 requests	MaxSpareServers: 256	MaxRequestsPerProcess: 200	Memory: 4 GB
	MaxClients: 256	IdleTimeout: 600	
	MaxRequestsPerChild: 0	BusyTimeout: 300	
		ErrorScanInterval: 60	

Refer "https://httpd.apache.org/mod_fcgid/mod/mod_fcgid.html" webpage for more details about "mod_fcgid" configuration options

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

Log Files for the AFP2web Server

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

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.

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

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.

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

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

Testing the AFP2web Server's Get Status Service:

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

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

- 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

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`.

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

URL

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

Request Parameters

	Parameter Name	Value
Input Document Parameters		
	InputBuffer	Content of file to be transformed
	InputUrls	<div>Input file to be transformed and can be any one of the following<ul style="list-style-type: none">• One or more input file names separated by comma.<div>These files must be accessible by AFP2web Server.</div><ul style="list-style-type: none">• One or more input URLs separated by comma.<div>Only following URL protocol is supported<ul style="list-style-type: none">• <code>http://</code></div></div>

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

Output Document Parameters		
	OutputMode	Buffer File Buffer: Default. Transformed document sent back as HTML with proper mime type File: Transformed document(s) will be written to a output directory specified by a A2W Parameter "OutputFilePath". This path must be accessible/writable from A2W Server
A2W Parameters		
	AFP2web parameters passed as Parameter=Value.	Refer AFP2web user guide to get parameter names
Server Parameters		
	UserAction	Must be "A2WTransform"
	transform_logging	on off. Turns on/off logging of Transformation. Default is off.

Response

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

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

Response when OutputMode is "File"

- Output Documents are stored on path specified in "OutputFilePath" parameter.
- After process is over, following HTTP Response will be sent to the caller either as HTML

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

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

Asynchronous Transformation Service

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

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

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.

URL

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

Request Parameters

	Parameter Name	Value
Input Document Parameters		
	InputUrls	<p>Input file to be transformed and can be any one of the following</p> <ul style="list-style-type: none">• One or more input file names separated by comma. <div>These files must be accessible by AFP2web Server.</div> <ul style="list-style-type: none">• One or more input URLs separated by comma. <div>Only following URL protocol is supported</div> <ul style="list-style-type: none">• <code>http://</code>
Output Document Parameters		
	OutputMode	<p>Must be File. File: Transformed document(s) will be written to a output directory specified by a A2W Parameter "OutputFilePath". This path must be accessible/writable from AFP2web Server</p>
	OutputFilePath	<p>If given, output files will be written in given output path.</p> <p>If not given, output files will be written in output path specified in afp2web.ini file</p>
A2W Parameters		
	AFP2web parameters passed as Parameter=Value.	Refer AFP2web user guide to get parameter names
Server Parameters		
	JobID	Unique Job Id for the asynchronous request
	UserAction	Must be "A2WTransform"
	transform_logging	on/off. Turns on/off logging of Transformation. Default is off.

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.

Get Status Service

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

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

URL

GET method: `http://<server ip>[:<service port>]/services/jobstatus?UserAction=getStatus&JobId=<JobId>`

Request Parameters

Parameter Name	Value
JobID	Unique Job ID. It is the same Job ID passed to the "asynctransform" request for which status is requested.
UserAction	Must be "getStatus"
transform_logging	on off. Turns on/off logging of Transformation. Default is off.

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>

Version Service

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

URL

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

Request Parameters

Parameter Name	Value
UserAction	Must be "VersionInfo"
transform_version	on off
transform_version_all	on off
a2w_version	on off
a2w_version_all	on off

Response

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

- Response in case of success

Version information returned as HTTP response

- Response in case of error

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

USID Service

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

URL

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

Request Parameters

Parameter Name	Value
transform_logging	on off (Optional), Turns on/off logging of Transformation. Default is off.

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>

AFP2web Server Additional Information

AFP2web Server Frequently Asked Questions

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`

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
```

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
```