



OXSEED Transformation Server Version 2.x User Guide and Reference

**OXSEED Transformation Server Version 2.x User Guide
and Reference**

Date: 2015-07-29

Maas Holding GmbH

Copyright © 2015 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

1	Preface	6
1.1	About the OXSEED Transformation Server User's Guide	6
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	6
1.2	Important Notices.....	7
1.2.1	Test thoroughly!	7
1.3	History.....	7
2	Introduction.....	9
3	User's Guide and Reference	10
3.1	Prerequisites	10
3.1.1	System Prerequisites	10
3.1.2	Software Prerequisites.....	10
3.2	Installing the OXSEED Transformation Server	13
3.2.1	Installing OTS	13
3.3	Configuring the OXSEED Transformation Server.....	15
3.3.1	Configuration	15
3.4	Plugins.....	15
3.4.1	oxsnps-afp2any.....	15
3.4.2	oxsnps-html2any	17
3.4.3	oxsnps-msoffice2any.....	18
3.4.4	oxsnps-pdf2pdf	21
3.4.5	oxsnps-soap	23
3.5	Starting / Stopping the OXSEED Transformation Server	36
3.6	About Splitting OXSEED Transformation Server Log Files	37
3.7	Testing the OXSEED Transformation Server.....	38
3.8	Updating the OXSEED Transformation Server and Plugins	39
4	List of public URLs	41
	List of Figures.....	42
	List of Tables	43

1 Preface

This preface offers

- a short introduction to the OXSEED Transformation Server (OTS)
- 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 OXSEED Transformation Server

1.1 About the OXSEED Transformation Server User's Guide

The OXSEED Transformation Server (OTS) is an unified server meant to meet all transformation requirements. It accepts transformation requests as HTTP or SOAP. Transformation services can be plugged in OTS.

1.1.1 Who Should Read this Document

This document is intended for all users of OTS. To use OTS, it is sufficient to set up its environment and the configuration files. Users who need to customize OTS will find information on how to do this in forthcoming chapters.

1.1.2 What's in this Document

This document is organized into the following chapters:

- This preface with an overview of the new features of OXSEED Transformation Server Version 2.x and some important notices
- An Introduction to the OXSEED Transformation Server, its functionality and possible application scenarios
- The OXSEED Transformation Server User's Guide, which includes information about installing and configuring it for your application, general information on how to run it, and how to use basic processing features
- ...

1.1.3 Where to Find Additional Information

For more information about

- Node.js: refer to <http://nodejs.org>
- Haraka SMTP Server: refer to <http://haraka.github.io>
- Apache HTTP Server: refer to <http://httpd.apache.org>
- JRE 7: refer to <http://www.oracle.com/technetwork/java/javase/7u21-relnotes-1932873.html#serverjre>

1.1.3.1 Whom to Contact

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

Maas Holding GmbH offers professional services and consulting related to the development and deployment of OXSEED Transformation Server-based solutions.

For more information, please visit our Web site at <http://www.oxseed.com/>

or contact us via: E-mail: afp2web@oxseed.de

1.2 Important Notices

Please use the Oxseed Transformation Server and components with great care. Incorrect settings can affect the transformation results. In the worst case, documents might become useless for archiving purposes.

We therefore recommend:

1.2.1 Test thoroughly!

You should thoroughly test the run time parameters and resources against all of your transformation scenarios.

1.3 History

Version	Date	Description
2.0.2	14.07.2015	Migrated plugins to oxsnps v2.0.2
2.0.0.1u1	30.09.2014	Added information about plugin installation and configuration. Modified OTS installation steps
2.0.0.0	18.09.2014	Added office2any, pdf2securepdf, pdf2mergepdf transformation plugins
1.3.5u18	06.08.2014	Added Update to V1.3.5u18 Section Added the List of available URLs Chapter
1.3.5u13	18.07.2014	Added Update to V1.3.5u13 Section
1.3.5u10	10.07.2014	Added URL <code>http://<server>:<port>/explore/segmentation</code> Added archivConf XML to JSON conversion
1.3.5u4	07.07.2014	Added Update to V1.3.5u4 Section Added forever module
1.3.5u3	03.07.2014	Added Update to V1.3.5u3 Section
1.3.5u2	30.06.2014	Added OTS port to OTS Configuration Added Update to V1.3.5u2 Section
1.3.5u1	26.06.2014	Added Update to V1.3.5u1 Section
1.3.5	25.06.2014	Added Update to V1.3.5 Section
1.3.4	10.06.2014	Installation Issues Added URL <code>http://<server>:<port>/version</code> Added URL <code>http://<server>:<port>/log</code> Added URL <code>http://<server>:<port>/doc</code> Added URL <code>http://<server>:<port>/segmentation/<infoType></code>
1.3.3	30.05.2014	Initial Release

1.3.2 alpha	18.03.2014	Alpha version delivered
----------------	------------	-------------------------

2 Introduction

Oxseed Transformation Server (OTS) is an unified server meant to meet all transformation requirements. To do so any transformation request is passed as SOAP, JSON, HTML through HTTP to OTS. The transformation will in turn be processed by OTS and appropriate actions will be taken to transform and return the result of the transformation.

Result will be, based on the request, either:

- sent back as SOAP, JSON, HTML response through HTTP
- stored on a file system. In that case, some status will be sent back as SOAP, JSON, HTML response through HTTP

3 User's Guide and Reference

3.1 Prerequisites

List of Prerequisites that are required for the installation of the OXSEED Transformation Server

- System Prerequisites
- Software Prerequisites

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

3.1.1 System Prerequisites

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

Description	Value
Operating System	Ubuntu 12.04.4, CentOS 7
Architecture	64 bit
Number of Cores	≥ 2
Memory	≥ 2 GB

3.1.2 Software Prerequisites

This chapter describes the list of the software packages that have to be installed in order to run OTS.

- node.js
- Java Development Kit (JDK) 7
- python 2.x

3.1.2.1 Installation of the node.js

Node.js is a platform for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

To install node.js on Ubuntu execute the following commands in the specified order.

Table 1 node.js Installation

```
sudo apt-get update
sudo apt-get install -y python-software-properties python g++ make
sudo add-apt-repository ppa:chris-lea/node.js
sudo apt-get update
sudo apt-get install nodejs
```

It installs node.js on Ubuntu.

To check which node.js version has been installed, execute the following command.

Table 2 node.js Version

```
node -v
```

It should return a version equal or greater than 0.10.26.

Also check that npm has been installed by executing the following command.

Table 3 npm Version

```
npm -v
```

It should return a version equal or greater than 1.4.3.

3.1.2.2 Installation of the JDK 7

Download and install the Oracle JDK 7 package from from

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Table 4 Installation of the JDK 7

```
sudo mkdir -p /usr/local/java
cd ~/Downloads
sudo cp jdk-7u67-linux-x64.tar.gz /usr/local/java/
cd /usr/local/java
sudo chmod a+x jdk-7u67-linux-x64.tar.gz
sudo tar xvzf jdk-7u67-linux-x64.tar.gz
```

Delete the tar file

```
sudo rm jdk-7u67-linux-x64.tar.gz
```

Set up the Java environment

Edit your profile. Add the the following entries at the end of your profile

Table 5 Edit /etc/profile

```
sudo nano /etc/profile

# Set the java environment
JAVA_HOME=/usr/local/java/jdk1.7.0_67
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export PATH
```

Inform Ubuntu where your Oracle Java JRE is located. This command will tell the system that the new Oracle Java version is available for use.

```
sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/java/jdk1.7.0_67/jre/bin/java"
```

Inform Ubuntu that Oracle Java JRE must be the default Java. This command will set the java run-time environment for the system.

```
sudo update-alternatives --set java /usr/local/java/jdk1.7.0_67/jre/bin/java
```

Check whether the Oracle JDK 7 was installed correctly. To do so, run the following command.

```
java -version
```

It should return:

```
java version "1.7.0_67"
Java(TM) SE Runtime Environment (build 1.7.0_67-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)
```

Once rebooted, Ubuntu will be fully configured for running Java programs.

3.1.2.3 Installation of python 2.x

OXSEED Transformation Server (OTS) needs python version 2.x.

To install python version 2.x on Ubuntu execute the following commands in the specified order.

Table 6 Installation of python 2.x

```
sudo apt-get update
sudo apt-get install pythonv2.7
```

It installs pythonv2.7 on Ubuntu.

To check which python version has been installed, execute the following command.

```
python --version
```

It should return a version 2.7.x as given below

```
Python 2.7.3
```

OTS requires python 2.x not python 3.x in order to install java node.js modules.

3.2 Installing the OXSEED Transformation Server

This section describes the OXSEED Transformation Server Installation.

In that document, the /opt/OTS directory is used as default installation directory if not otherwise specified. Moreover this document assumes that the downloaded files reside in the ~/Downloads directory.

3.2.1 Installing OTS

In order to install OXSEED Transformation Server (OTS), first download the OTS package and then execute the following commands.

Table 7 OTS installation

```
sudo mkdir -p /opt/OTS
sudo cp ~/Downloads/OTSServer_V2.x.x_linux_x64.tgz /opt/OTS
cd /opt/OTS
sudo chmod a+x OTSServer_V2.x.x_linux_x64.tgz
sudo tar xvf OTSServer_V2.x.x_linux_x64.tgz
```

This will create the following directories within /opt/OTS:

Directory Name	Remarks
----------------	---------

conf	Contains the OTS configuration.
doc	Contains the OTS documentation.
lib	Contains jar files needed to call msoffice-converter and pdf2pdf converter java services.
log	Directory to save log files.
node_modules	Contains OTS Plugins.
public	Static files to be served by OTS

Once the installation is over, delete the tar file

```
sudo rm OTSServer_V2.x.x_linux_x64.tgz
```

OTS requires additional node.js modules. In order to install them run the following command. This would install additional node.js modules in 'node_modules' directory.

Table 8 Install the additional node.js modules

```
npm install
```

OTS also requires PM2 module that helps to keep applications alive for ever, to reload them without downtime.

Table 9 Install PM2 node.js module

```
npm install pm2 -g --unsafe-perm
```

PM2 Information

Refer to <https://github.com/Unitech/PM2/> get more information about PM2

3.2.1.1 wkhtmltopdf Installation

HTML to PDF/TIF transformations require the wkhtmltopdf tool. In order to install wkhtmltopdf, do the following steps.

- Download wkhtmltox-0.12.2.1 into ~/Downloads for your operating system from <http://wkhtmltopdf.org/downloads.html>
- Install wkhtmltopdf

Table 10 Installing wkhtmltox on Ubuntu

```
sudo dpkg -i ~/Downloads/wkhtmltox-0.12.2.1_linux-precise-amd64.deb
```

3.3 Configuring the OXSEED Transformation Server

This section describes the OXSEED Transformation Server Configuration.

The OXSEED Transformation Server Configuration is maintained in the `/opt/OTS/conf/server.json` file. This file is a JSON file. It contains all the OTS settings.

In order to run the OXSEED Transformation Server you will need to configure `conf/server.json` file with proper values.

3.3.1 Configuration

Define the port to listen to, the logging level (should be INFO in production), set the name of the log, temp, upload and backup directories. Default log directory is `/var/log/nps`. Default log file is `/var/log/nps/npsServer.log`. Default temp directory is `/tmp/nps`. The upload directory is used to store the releases to be deployed. The backup directory is used to store the backup that will be done before deploying the new release. Available logging levels are DEBUG, INFO and ERROR.

Table 11 npsServer Configuration

```
{
  "npsServer":      "OTS-v2.0.2",
  "desc":           "OXSEED Transformation Server",
  "port":           "OXSNPS Server Port",
  "port":           80,
  "logLevel":       "Available Log Levels are DEBUG, INFO and ERROR",
  "logLevel":       "INFO",
  "logDir":         "/var/log/nps",
  "tempDir":        "/tmp/nps",
  "uploadDir":      "uploads",
  "backupDir":      "backups"
}
```

3.4 Plugins

This section describes the plugins available to the OXSEED Transformation Server and their configuration.

3.4.1 oxsnps-afp2any

The `oxsnps-afp2any` plugin allows OTS to access the transformations provided by the AFP2web Server. Its configuration is maintained in `/opt/OTS/node-modules/oxsnps-afp2any/conf/oxsnps-afp2any.json`.

You will mostly have to set in that configuration the AFP2web Server IP Address and its port.

The `oxsnps-afp2any.json` is a JSON file. It contains the following properties.

Property Name	Description
---------------	-------------

module	Name of the plugin.
desc	Plugin description.
longDesc	Plugin long description.
enable	Enable Flag. Values: on off. Default is 'off'.
logLevel	Plugin Log Level. Values: DEBUG INFO ERROR. Default is 'INFO'.
props	JSON object specifying plugin initialization parameters
routes	Array of Routes. A Route describes the URI of the service ("path" property), the HTTP method to use for calling the service ("method" property) and the name of the service ("service " property) to which the request must be routed to.
name	AFP2web Server IP Address.
port	AFP2web Server Port.
logging	AFP2web Server Logging. Values: On Off All. On: Turn on Server Logging. Off: Turn off Server Logging. All: Turn on Server and AFP2web Logging.

Table 12 oxsnps-afp2any.json

<pre> { "module": "Name of the plugin", "module": "oxsnps-afp2any", "desc": "OXSNPS afp2any Plugin", "longDesc": "OXSEED NodeJS Server afp2any Plugin", "enable": "on", "logLevel": "Available Log Levels are DEBUG, INFO and ERROR", "logLevel": "INFO", "routes": [{ "path": "/services/afp2any", "method": "get", "service": "main" }, { "path": "/services/afp2any/version", "method": "get", "service": "version" }, { "path": "/services/afp2any/transform", "method": "post", "service": "syncTransform" }] } </pre>

```

    },
    {
      "path":      "/services/afp2any/asynctransform",
      "method":    "post",
      "service":    "asyncTransform"
    },
    {
      "path":      "/services/afp2any/status",
      "method":    "post",
      "service":    "getStatus"
    }
  ],
  "name":      "AFP2web Server IP Address",
  "name":      "localhost",
  "port":      "AFP2web Server Port",
  "port":      8080,
  "logging":   "Turn AFP2web Server Logging On/Off/All",
  "logging":   "Off"
}

```

For further details about the AFP2web Server refer to AFP2web Server User Guide and Reference.

By default, both OTS and AFP2web Server listens on port 80 for incoming requests. In order to avoid the conflict, AFP2web Server must be configured to listen on different port for example 8080. Refer to the 'Port configuration' Section in AFP2web Server User Guide to get information on using different port for AFP2web Server.

3.4.2 oxsnps-html2any

The oxsnps-html2any plugin provides html to pdf and tif transformations. Its configuration is maintained in /opt/OTS/oxsnps-html2any/conf/oxsnps-html2any.json.

3.4.2.1 oxsnps-html2any.json

The oxsnps-html2any.json file describes the plugin configuration. It contains following properties.

Property Name	Description
module	Name of the plugin
desc	Plugin description
longDesc	Plugin long description
enable	Enable Flag. Values: on off. Default is 'off'
logLevel	Plugin Log Level. Values: DEBUG INFO ERROR. Default is 'INFO'
props	JSON object specifying plugin initialization parameters

routes	Array of Routes. A Route describes the URI of the service ("path" property), the HTTP method to use for calling the service ("method" property) and the name of the service ("service " property) to which the request must be routed to.
--------	---

Table 13 oxsnps-html2any.json

<pre> { "module": "Name of the plugin", "module": "oxsnps-html2any", "desc": "OXSNPS html2any Plugin", "longDesc": "OXSEED NodeJS Server html2any Plugin", "enable": "on", "logLevel": "Available Log Levels are DEBUG, INFO and ERROR", "logLevel": "info", "routes": [{ "path": "/services/<plugin name>/<service name>", "path": "/services/html2any/version", "method": "get", "service": "version" }] }</pre>
--

3.4.3 oxsnps-msoffice2any

The oxsnps-msoffice2any plugin provides transformations of Microsoft Office documents. Its configuration is maintained in following files.

1. /opt/OTS/node-modules/oxsnps-msoffice2any/conf/oxsnps-msoffice2any.json: It is a JSON file describing the plugin configuration
2. /opt/OTS/node-modules/oxsnps-msoffice2any/conf/dcom.json: It is a JSON file describing the DCOM Server configuration
3. /opt/OTS/node-modules/oxsnps-msoffice2any/conf/log4j.xml: It contains log configuration for the java services.

3.4.3.1 oxsnps-msoffice2any.json

The oxsnps-msoffice2any.json file describes the plugin configuration. It contains following properties.

Property Name	Description
module	Name of the plugin
desc	Plugin description
longDesc	Plugin long description
enable	Enable Flag. Values: on off. Default is 'off'

logLevel	Plugin Log Level. Values: DEBUG INFO ERROR. Default is 'INFO'
props	JSON object specifying plugin initialization parameters
routes	Array of Routes. A Route describes the URI of the service ("path" property), the HTTP method to use for calling the service ("method" property) and the name of the service ("service " property) to which the request must be routed to.

Table 14 oxsnps-msoffice2any.json

<pre> { "module": "Name of the plugin", "module": "oxsnps-msoffice2any", "desc": "OXSNPS msoffice2any Plugin", "longDesc": "OXSEED NodeJS Server msoffice2any Plugin", "enable": "on", "logLevel": "Available Log Levels are DEBUG, INFO and ERROR", "logLevel": "info", "props": { "confile": "Plugin Configuration file. Must exist in the plugins dir", "confile": "conf/dcom.json" }, "routes": [{ "path": "/services/<plugin name>/<service name>", "path": "/services/msoffice2any/version", "method": "get", "service": "version" }, { "path": "/services/msoffice2any/transform", "method": "post", "service": "transform" }] }</pre>

3.4.3.2 dcom.json

The dcom.json file describes the configuration of the DCOM server. It contains following properties.

Property Name	Description
PDFCreatorApplication	Name of the PDF Creator application.
PDFProducer	Tool or library used to generate PDF file.
dcomDomain	Domain name or host name of the DCOM server
dcomHost	IP Address or name of the DCOM server
dcomUser	User that has permissions to access the DCOM server
dcomPassword	Password of DCOM user
dcomShare	URL to access the DCOM share directory

	<p>Example 1: Share directory if DCOM server runs in different system than OTS server</p> <ul style="list-style-type: none"> • "dcomShare": "smb://172.15.0.28/tmp/" <p>Example 2: Share directory if OTS runs on Windows and DCOM server runs locally on the same system as OTS server</p> <ul style="list-style-type: none"> • "dcomShare": "file:///c:/OCTP1.2/temp/"
numberWordAppInPool	Number of word applications to be started
numberExcelAppInPool	Number of excel applications to be started
horizontalPrintQuality	Horizontal print quality of the TIFF transformation
verticalPrintQuality	Vertical print quality of the TIFF transformation
timeout	Transformation Time out in milliseconds

Table 15 dcom.json

<pre>{ "PDFCreatorApplication": "OTS v2.0.0.2", "PDFProducer": "Maas PDF Library v3.4", "DCOM": { "dcomDomain": server>, "dcomHost": server>, "dcomUser": access the DCOM server>, "dcomPassword": directory>, "dcomShare": started>, "numberWordAppInPool": started, "numberExcelAppInPool": transformation>, "horizontalPrintQuality": transformation>, "verticalPrintQuality": milliseconds> } }</pre>	<p><Domain name or host name of the DCOM</p> <p><IP Address or name of the DCOM</p> <p><User that has permissions to</p> <p><Password of DCOM user>,</p> <p><URL to access the DCOM share</p> <p><Number of word applications to be</p> <p><Number of excel applications to be</p> <p><Horizontal print quality of the TIFF</p> <p><Vertical print quality of the TIFF</p> <p><Transformation Time out in</p>
---	--

For further details about the DCOM Server refer to the Configuring_DCOM_XP_2003.pdf and Office2Any Server SetUp-23 Sep 2008.pdf documents in the 'doc' directory..

3.4.3.3 log4j.xml:

The log4j.xml file describes the log configuration of the java services. It contains following properties. The name of the log file for java services is specified in the "File" <param> element of the 'FILE' <appender> as shown below.

Table 16 log4j.xml

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/"
debug="false">
    .....
    <appender name="FILE" class="org.apache.log4j.FileAppender">
        <!-- 'threshold' defines log level. Allowed values are
debug|info|warn|error -->
        <param name="threshold" value="debug"/>
        <param name="File" value="/var/log/nps/javaServices.log"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSS}
[%-5p] %-25.25c{1} %4L - %m%n"/>
        </layout>
    </appender>
    .....
</log4j:configuration>
```

3.4.4 oxsnps-pdf2pdf

The oxsnps-pdf2pdf plugin provides transformations to:

- secure a PDF Document
- merge and secure the merged PDF document

The oxsnps-pdf2pdf plugin configuration is maintained in the following files.

1. /opt/OTS/oxsnps-pdf2pdf/conf/oxsnps-pdf2pdf.json: It is a JSON file describing the plugin configuration
2. /opt/OTS/oxsnps-pdf2pdf/conf//log4j.xml: It contains log configuration for the java services

3.4.4.1 oxsnps-pdf2pdf.json

The oxsnps-pdf2pdf.json file describes the plugin configuration. It contains following properties.

Property Name	Description
module	Name of the plugin
desc	Plugin description
longDesc	Plugin long description
enable	Enable Flag. Values: on off. Default is 'off'
logLevel	Plugin Log Level. Values: DEBUG INFO ERROR. Default is 'INFO'
props	JSON object specifying plugin initialization parameters

routes	<p>Array of Routes. A Route describes the URI of the service ("path" property), the HTTP method to use for calling the service ("method" property)</p> <p>and the name of the service ("service " property) to which the request must be routed to.</p>
--------	---

Table 17 oxsnps-pdf2pdf.json

<pre> { "module": "Name of the plugin", "module": "oxsnps-pdf2pdf", "desc": "OXSNPS pdf2pdf Plugin", "longDesc": "OXSEED NodeJS Server pdf2pdf Plugin", "enable": "on", "logLevel": "Available Log Levels are DEBUG, INFO and ERROR", "logLevel": "info", "routes": [{ "path": "/services/<plugin name>/<service name>", "path": "/services/pdf2pdf/version", "method": "get", "service": "version" }, { "path": "/services/pdf2pdf/transform", "method": "post", "service": "transform" }, { "path": "/services/pdf2pdf/merge", "method": "post", "service": "merge" }] }</pre>
--

3.4.4.2 log4j.xml:

The log4j.xml file describes the log configuration of the java services. It contains following properties. The name of the log file for java services is specified in the "File" <param> element of the 'FILE' <appender> as shown below.

Table 18 log4j.xml

<pre> <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/" debug="false"> <appender name="FILE" class="org.apache.log4j.FileAppender"> <!-- 'threshold' defines log level. Allowed values are debug info warn error --> <param name="threshold" value="debug"/> <param name="File" value="/var/log/nps/javaServices.log"/> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSS} [%-5p] %-25.25c{1} %4L - %m%n"/> </layout> </appender> </log4j:configuration></pre>
--

3.4.5 oxsnps-soap

The oxsnps-soap plugin adds SOAP support to OTS. It parses SOAP requests, delegates them to the proper functional plugin like oxsnps-msoffice2any, oxsnps-afp2any or oxsnps-pdf2pdf and sends the SOAP response back to the caller. Its configuration is maintained in /opt/OTS/node-modules/oxsnps-soap/conf/oxsnps-soap.json. The oxsnps-soap.json is a JSON file. It contains the following properties.

Property Name	Description
module	Name of the plugin
desc	Plugin description
longDesc	Plugin long description
enable	Enable Flag. Values: on off. Default is 'off'
logLevel	Plugin Log Level. Values: DEBUG INFO ERROR. Default is 'INFO'
props	JSON object specifying plugin initialization parameters
routes	Array of Routes. A Route describes the URI of the service ("path" property), the HTTP method to use for calling the service ("method" property) and the name of the service ("service" property) to which the request must be routed to.

Table 19 oxsnps-soap.json

<pre>{ "module": "Name of the plugin module", "module": "oxsnps-soap", "desc": "OXSNPS soap Plugin", "longDesc": "OXSEED NodeJS Server soap Plugin", "enable": "on", "logLevel": "Available Log Levels are DEBUG, INFO and ERROR", "logLevel": "INFO" "routes": [{ "path": "This path is maintained since it has been used in Sanlam OCTP for SOAP sync.transformations", "path": "/TransformService", "method": "post", "service": "syncTransform" }, { "path": "/services/soap/transform", "method": "post", "service": "syncTransform" }, { "path": "/services/soap/asynctransform", "method": "post", "service": "asyncTransform" }], }</pre>

```

        {
            "path":      "/services/soap/status",
            "method":    "post",
            "service":   "status"
        }
    ]
}

```

3.4.5.1 SOAP Interface

This chapter describes the SOAP interface of the OXSEED Transformation Server.

Transformation WSDL

Here is the SOAP Transformation WSDL

Table 20 Service WSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<wSDL:definitions
  xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
  xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
  xmlns:tns="http://oxseed.com/services/"
  xmlns:wSDLsoap="http://schemas.xmlsoap.org/wSDL/soap/"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://oxseed.com/services/">
  <wSDL:types>
    <xsd:schema
      attributeFormDefault="qualified"
      elementFormDefault="unqualified"
      targetNamespace="http://oxseed.com/services/">
      <xsd:simpleType name="ResponseStatus">
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Done"/>
          <xsd:enumeration value="Error"/>
          <xsd:enumeration value="Working"/>
        </xsd:restriction>
      </xsd:simpleType>

      <xsd:element name="ResponseMessage">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="responseData"
              type="xsd:base64Binary"/>
            <xsd:element name="responseStatus"
              type="tns:ResponseStatus"/>
            <xsd:element name="responseStatusDescription"
              type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:complexType name="InputObject">
        <xsd:sequence>
          <xsd:element name="source" type="xsd:string"/>
          <xsd:element name="ownerpw" type="xsd:string"/>
          <xsd:element name="userpw" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wSDL:types>
</wSDL:definitions>

```



```

        <xsd:complexType name="OutputObject">
            <xsd:sequence>
                <xsd:element name="type" type="xsd:string"/>
                <xsd:element name="ownerpw" type="xsd:string"/>
                <xsd:element name="userpw" type="xsd:string"/>
                <xsd:element name="flags" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>

        <xsd:element name="RequestMergeMessage">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" minOccurs="1"
name="inputList" type="tns:InputObject"/>
                    <xsd:element name="output" type="tns:OutputObject"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

        <xsd:complexType name="InputObjectTransform">
            <xsd:complexContent>
                <xsd:extension base="tns:InputObject">
                    <xsd:sequence>
                        <xsd:element name="document"
type="xsd:base64Binary"/>
                        <xsd:element name="type" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>

        <xsd:element name="RequestTransformMessage">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="input"
type="tns:InputObjectTransform"/>
                    <xsd:element name="output" type="tns:OutputObject"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:schema>
</wsdl:types>

<wsdl:message name="requestMergeMsg">
    <wsdl:part element="tns:RequestMergeMessage" name="request">
    </wsdl:part>
</wsdl:message>

<wsdl:message name="requestTransformMsg">
    <wsdl:part element="tns:RequestTransformMessage" name="request">
    </wsdl:part>
</wsdl:message>

<wsdl:message name="responseMsg">
    <wsdl:part element="tns:ResponseMessage" name="result">
    </wsdl:part>
</wsdl:message>

<wsdl:portType name="TransformInterface">
    <wsdl:operation name="merge">
        <wsdl:input message="tns:requestMergeMsg">
        </wsdl:input>
        <wsdl:output message="tns:responseMsg">
        </wsdl:output>
    </wsdl:operation>
</wsdl:portType>

```

```

    </wsdl:operation>
    <wsdl:operation name="transform">
      <wsdl:input message="tns:requestTransformMsg">
    </wsdl:input>
      <wsdl:output message="tns:responseMsg">
    </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="TransformServicePortBinding" type="tns:TransformInterface">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="merge">
      <soap:operation soapAction=""/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="transform">
      <soap:operation soapAction=""/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="TransformService">
    <wsdl:port binding="tns:TransformServicePortBinding"
name="TransformServicePort">
      <soap:address location="http://0.0.0.0:8193/TransformService/">
    </soap:address>
      <soap:address location="refer service urls below"/>
    </soap:address>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Following chapters describe some sample SOAP requests and their response for different transformations.

SOAP Request

In order to run the AFP2Any transformation, such a SOAP Request has to be sent to OTS:

Table 21 AFP2Any Request

```

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:transform xmlns:ns1="http://oxseed.com/services/">
      <ns1:RequestTransformMessage>
        <input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns1:InputObjectTransform">

```

```

<document>L3Zhci9saWlvYXBhY2hlMi9kYXRhL3NhbnBzZXMvYmlnZmlsZXRLc3QuYWZw</document>

    <type>afp|bmp|pcx|png|jpg|tif</type>
  </input>
  <output>
    <type>OutputType</type>
    <ownerpw>oxdoxd</ownerpw>
    <userpw>oxd</userpw>

  </output>
</ns1:RequestTransformMessage>
</ns1:transform>
</soapenv:Body>
</soapenv:Envelope>

```

where

- <input><document> is data of the input pdf document, encoded in base64.
- <input><type> is type of input document. Refer to "Input/Output Parameter Types" table below.
- <output><type> is type of output document. Refer to "Input/Output Parameter Types" table below.
- <output><ownerpw> is owner password of the output pdf document. Applicable only for PDF output type.
- <output><userpw> is user password of the output pdf document. Applicable only for PDF output type.

Input/Output Parameter Types

"any" input type means one of the following input types AFP, PTOCA(MO:DCA IS1/2), BMP, PCX, PNG, JPG, TIF

Conversion	Input Type	Output Type
any2afp (Color)	any	afp
any2pdf (Color)	any	pdf
any2tif (B&W)	any	tif_bw
pdf2afp (Color)	pdf	afp
pdf2afp (Grayscale)	pdf	afp_gr
pdf2afp (B&W)	pdf	afp_bw
pdf2tif (Color)	pdf	tif
pdf2tif (B&W)	pdf	tif_bw
gif2pdf (Color)	gif	pdf
gif2tif_bw (B&W)	gif	tif_bw
gif2tif (Color)	gif	tif

gif2afp (Color)	gif	afp
txt2pdf (Color)	txt	pdf

SOAP Response

The output should look as follows.

Table 22 AFP2Any Response

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header/>
  <soap:Body>
    <ox:ResponseMessage xmlns:ox="http://oxseed.com/services/">
      <responseData>JVBERi0xLjIKJdTBeIKMS</responseData>
      <responseStatus>Done|Error</responseStatus>
      <responseStatusDescription/>
    </ox:ResponseMessage>
  </soap:Body>
</soap:Envelope>
```

where

- <responseData> contains the output document data encoded in base64, in case of successful transformation.
- <responseStatus> contains the status of transformation. Possible values are 'Done' or 'Error'.
- <responseStatusDescription> contains the error description.

SOAP Request

In order to run the Office2Any transformation, such a SOAP Request has to be sent to OTS:

Table 23 Office2any Request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:transform xmlns:ns1="http://oxseed.com/services/">
      <ns1:RequestTransformMessage>
        <input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="ns1:InputObjectTransform">
          <ownerpw>mhtmht</ownerpw>
          <userpw>mht</userpw>

          <document>A3ZYci9saWIvYXBhY2hlMi9kYXRhL3NhbXBsZXMvYmlnZmlsZXRLc3QuYWZw</document>

          <type>InputType</type>
        </input>
        <output>
          <type>OutputType</type>
        </output>
      </ns1:RequestTransformMessage>
    </ns1:transform>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        </output>
      </ns1:RequestTransformMessage>
    </ns1:transform>
  </soapenv:Body>
</soapenv:Envelope>
<soapenv:Header/>

```

where

- <input><ownerpw> is owner password of the input document.
- <input><userpw> is user password of the input document.
- <input><document> is data of the input document, encoded in base64.
- <input><type> is type of input document. Refer to "Input/Output Parameter Types" table below.
- <output><type> is type of output document. Refer to "Input/Output Parameter Types" table below.

Input/Output Parameter Types

Conversion	Input Type	Output Type
Ms Office word to PDF Ms Office word 2007 to PDF	word	pdf
Html to PDF	mht	pdf
"Html information presented like table format" to PDF	html	pdf
RTF to PDF	rtf	pdf
Ms Excel to PDF Ms Excel 2007 to PDF	excel	pdf
Ms Office word to TIF Ms Office word 2007 to TIF	word	tif
Ms Office word to TIF (BW) Ms Office word 2007 to TIF (BW)	word	tif_bw
RTF to TIFF	rtf	tif
RTF to TIFF (BW)	rtf	tif_bw
Ms Excel to TIF Ms Excel 2007 to TIF	excel	tif
Ms Excel to TIF (BW)	excel	tif_bw

SOAP Response

The output should look as follows:

Table 24 Office2any Response

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header/>
  <soap:Body>
    <ox:ResponseMessage xmlns:ox="http://oxseed.com/services/">
      <responseData>Z3H7JVBERi0xLjIKJdTBeIKMS</responseData>
      <responseStatus>Done|Error</responseStatus>
      <responseStatusDescription/>
    </ox:ResponseMessage>
  </soap:Body>
</soap:Envelope>
```

where

- <responseData> contains the output document data encoded in base64, in case of successful transformation.
- <responseStatus> contains the status of transformation. Possible values are 'Done' or 'Error'.
- <responseStatusDescription> contains the error description.

SOAP Request

In order to run the PDF Secure transformation, such a SOAP Request has to be sent to OTS:

Table 25 pdf2securepdf Request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:transform xmlns:ns1="http://oxseed.com/services/">
      <ns1:RequestTransformMessage>
        <input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="ns1:InputObjectTransform">
          <ownerpw>mhtmht</ownerpw>
          <userpw>mht</userpw>

          <document>F3Thci9saWIvYXBhY2hlMi9kYXRhL3NhbXBsZXMvYmlnZmlsZXRXRlc3QuYWZw</document>

          <type>pdf</type>
        </input>
        <output>
          <type>pdf</type>
          <ownerpw>oxdoxd</ownerpw>
          <userpw>oxd</userpw>
        </output>
      </ns1:RequestTransformMessage>
    </ns1:transform>
  </soapenv:Body>
</soapenv:Envelope>
```

```

                                <flags>1|2|3|4|5|6</flags>
                                </output>
                                </ns1:RequestTransformMessage>
                                </ns1:transform>
                                </soapenv:Body>
                                </soapenv:Envelope>

```

where

- <input><ownerpw> is owner password of the input pdf document.
- <input><userpw> is user password of the input pdf document.
- <input><document> is data of the input pdf document, encoded in base64.
- <input><type> is always 'pdf'.
- <output><type> is always 'pdf'.
- <output><ownerpw> is owner password of the output pdf document.
- <output><userpw> is user password of the output pdf document.
- <output><flas> is permission flags applied on output pdf document. Values are separated by '|'. Refer 'Permission Flags' table below.

Permission Flags

Flag	Description
0	Disables all permissions
1	Allow low resolution printing
2	Allow changing the document
3	Allow content copying or extraction
4	Allow authoring comments and form fields
5	Allow form field fill-in or signing
6	Allow content accessibility
7	Allow document assembly
8	Allow high quality printing

SOAP Response

The output should looks as follow:

Table 26 pdf2securepdf Response

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header/>
  <soap:Body>
    <ox:ResponseMessage xmlns:ox="http://oxseed.com/services/">
      <responseData>M3k7JVBERi0xLjIKJdTbweIKMS</responseData>
      <responseStatus>Done|Error</responseStatus>
      <responseStatusDescription/>
    </ox:ResponseMessage>
  </soap:Body>
</soap:Envelope>
```

where

- <responseData> contains the output document data encoded in base64, in case of successful transformation.
- <responseStatus> contains the status of transformation. Possible values are 'Done' or 'Error'.
- <responseStatusDescription> contains the error description.

SOAP Request

In order to run the PDF merge transformation, such a SOAP Request has to be sent to OTS:

Table 27 MergePDF Request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header />
  <soapenv:Body>
    <ns1:merge xmlns:ns1="http://oxseed.com/services/">
      <ns1:RequestMergeMessage>
        <inputList>

          <source>/mnt/afp2webdata/test/customertest/sanlam/octp/testdaten/sample-secure.pdf</source>
            <ownerpw>mht</ownerpw>
            <userpw>mht</userpw>
          </inputList>
          <inputList>

          <source>/mnt/afp2webdata/test/customertest/sanlam/octp/testdaten/sample-secure.pdf</source>
            <ownerpw>mht</ownerpw>
            <userpw>mht</userpw>
          </inputList>
          <output>
            <type>mergedpdf</type>
            <ownerpw>oxdoxd</ownerpw>
            <userpw>oxd</userpw>
            <flags>1|2|3|4|5|6</flags>
          </output>
        </ns1:RequestMergeMessage>
      </ns1:merge>
```



```
</soapenv:Body>
</soapenv:Envelope>
```

where

- <inputList><source> is absolute path of input pdf file. This file must exist and accessible from OTS server.
- <inputList><ownerpw> is owner password of the input pdf document.
- <inputList><userpw> is user password of the input pdf document.
- <inputList><document> is data of the input pdf document.
- <output><type> is always 'mergedpdf'.
- <output><ownerpw> is owner password of the output pdf document.
- <output><userpw> is user password of the output pdf document.
- <output><fls> is permission flags applied on output pdf document. Values are separated by '|'. Refer 'Permission Flags' table below.

Permission Flags

Flag	Description
0	Disables all permissions
1	Allow low resolution printing
2	Allow changing the document
3	Allow content copying or extraction
4	Allow authoring comments and form fields
5	Allow form field fill-in or signing
6	Allow content accessibility
7	Allow document assembly
8	Allow high quality printing

SOAP Response

The output should look as follows:

Table 28 MergePDF Response

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header/>
  <soap:Body>
    <ox:ResponseMessage xmlns:ox="http://oxseed.com/services/">
      <responseData>JVBERi0xLjIKJdTbWeIKMS</responseData>
      <responseStatus>Status</responseStatus>
      <responseStatusDescription/>
    </ox:ResponseMessage>
  </soap:Body>
</soap:Envelope>
```

where

- <responseData> contains the output document data encoded in base64, in case of successful transformation.
- <responseStatus> contains the status of transformation. Possible values are 'Done' or 'Error'.
- <responseStatusDescription> contains the error description.

SOAP Request

In order to run the HTML2Any transformation, set the XML <input><type> tag to 'mht' and the XML <output><type> tag to 'pdf' resp. 'tif' as shown below:

Table 29 Office2any Request

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:transform xmlns:ns1="http://oxseed.com/services/">
      <ns1:RequestTransformMessage>
        <input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns1:InputObjectTransform">
          <document>A3ZYci9saWIvYXBhY2hlMi9kYXRhL3NhbXBsZXMvYmlnZmlsZXRXIc3QuYWZw</docum
ent>
          <type>mht</type>
        </input>
        <output>
          <type>pdf</type>
        </output>
      </ns1:RequestTransformMessage>
    </ns1:transform>
  </soapenv:Body>
</soapenv:Envelope>
<soapenv:Header/>
```

where

- `<input><document>` is data of the HTML input document, encoded in base64.
- `<input><type>` is the input document type. Refer to "Input/Output Parameter Types" table below.
- `<output><type>` is output document type. Refer to "Input/Output Parameter Types" table below.

Input/Output Parameter Types

Conversion	Input Type	Output Type
Html to PDF	mht	pdf
Html to TIFF	mht	tif

SOAP Response

The output should look as follow:

Table 30 Office2any Response

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header/>
  <soap:Body>
    <ox:ResponseMessage xmlns:ox="http://oxseed.com/services/">
      <responseData>Z3H7JVBERi0xLjIKJdTbweIKMS</responseData>
      <responseStatus>Done|Error</responseStatus>
      <responseStatusDescription/>
    </ox:ResponseMessage>
  </soap:Body>
</soap:Envelope>
```

where

- `<responseData>` contains the output document data encoded in base64, in case of successful transformation.
- `<responseStatus>` contains the status of transformation. Possible values are 'Done' or 'Error'.
- `<responseStatusDescription>` contains the error description in case of error.

Transformation Options

The `wkhtmltopdf` command line tool provides many transformation options for HTML to PDF generation. Refer to [wkhtmltopdf](#).

These options can be specified in "MHT2PDF" and "MHT2TIF" JSON objects in `/opt/OTS/conf/OTSScenarioConf.json` using 'lowerCamelCase' convention where the first letter of the entire word is lowercase, but subsequent first letters are uppercase. Refer to the example below.

```
"MHT2PDF": "HTML 2 PDF Conversion",
"MHT2PDF": {
```

```

    "props": "Options for PDF generation. Refer to
    http://madalgo.au.dk/~jakobt/wkhtmltoxdoc/wkhtmltopdf_0.10.0_rc2-doc.html.",
    "props": "Options are specified using lowerCamelCase convention where the
    first letter of the entire word is lowercase, but subsequent first letters are
    uppercase.",
    "props": "Example. If wkhtml2pdf tool option is --page-size, it should be
    specified as given below",
    "props": {
      "pageSize": "letter",
      "orientation": "Landscape"
    }
  }
}

```

3.5 Starting / Stopping the OXSEED Transformation Server

To start the OXSEED Transformation Server execute the following commands

Table 31 Starting OTS

```

cd /opt/OTS
pm2 start node_modules/oxsnps-core/server.js --name OTSV2

```

If you get the following output on the console, OTS has been started successfully.

```

root@adminuser-VirtualBox: /opt/OTS_Sanlam_V2.0.2
root@adminuser-VirtualBox:/opt/OTS_Sanlam_V2.0.2# pm2 start node_modules/oxsnps-core/server.js --name OTSV2
Starting PM2 daemon...
[PM2] Process node_modules/oxsnps-core/server.js launched

```

App name	id	mode	PID	status	restarted	uptime	memory	watching
OTSV2	0	fork	6864	online	0	0s	15.656 MB	unactivated

```

Use `pm2 desc[ribe] <id>` to get more details
root@adminuser-VirtualBox:/opt/OTS_Sanlam_V2.0.2#

```

You can also check the log file by running:

Table 32 Checking OTS Log

```
tail -f /var/log/nps/npsServer.log.
```

The output should look as follow.

```

root@adminuser-VirtualBox:/opt/OTS_Sanlam_V2.0.2# tail -f
/var/log/nps/npsServer.log

```

generic

2.90GHz

Hostname: adminuser-VirtualBox, linux x64 3.11.0-20-

Average Load: 0.208984375,0.11474609375,0.12841796875

Total Mem.: 5.83GB, Free Mem.: 4.15GB.

CPU 1, Model: Intel(R) Core(TM) i7-3520M CPU @

```

CPU 2, Model: Intel(R) Core(TM) i7-3520M CPU @
2.90GHz
Network Interfaces: eth3: 172.15.0.15
[2015-07-21 10:33:19.740] [INFO] oxsnps-segserver - Plugin oxsnps-segserver
v2.0.9 initialized
[2015-07-21 10:33:19.758] [INFO] oxsnps-pdf2pdf - Plugin oxsnps-pdf2pdf
v2.2.1 initialized
[2015-07-21 10:34:30.194] [INFO] oxsnps-msoffice2any - Plugin oxsnps-
msoffice2any v2.2.1 initialized
[2015-07-21 10:34:30.197] [INFO] npsServer - OXSEED Transformation Server v2.0.2
is listening on port 80

```

To stop the OXSEED Transformation Server, execute the following commands. For more information about the pm2 commands refer to <https://github.com/Unitech/PM2/>

```
pm2 kill
```

3.6 About Splitting OXSEED Transformation Server Log Files

To avoid to have to handle large log files, Linux provides the **logrotate** command that allows to split log files daily, monthly or yearly. To do so, create a `/etc/logrotate.d/ots.log` file and add the following configuration to it:

Table 33 OTS Logrotate Definition File

```

/var/log/nps/*.log {
    daily
    rotate 60
    missingok
    notifempty
    compress
    delaycompress
    sharedscripts
    copytruncate
    dateext
    dateformat %Y-%m-%d.
}

```

where

- `/var/log/nps/*.log` specifies which log files to rotate
- `daily` specifies to rotate the log files daily
- `rotate 60` specifies to keep up 60 old log files
- `missingok` specifies that if the log file doesn't exist it's not an issue
- `notifempty` specifies to ignore empty files
- `compress` specifies to compress the rotated log file with gzip
- `delaycompress` specifies to postpone the compression of the previous log file to the next rotation cycle

- `sharedscripts` specifies that the **prescript** and **postscript** scripts should run once only,
- `copytruncate` is needed for forever to work properly
- `dateext` specifies to add the date to filename
- `dateformat %Y-%m-%d` specifies the date format

To activate this configuration run the following command

Table 34 Activating OTS Logrotate Definition File

```
logrotate -s /var/log/logstatus /etc/logrotate.conf
```

For further details about the **logrotate** command, please refer to the links below:

1. http://linuxcommand.org/man_pages/logrotate8.html
2. <http://stackoverflow.com/questions/15231968/nodejs-forever-archive-logs>
3. <http://www.thegeekstuff.com/2010/07/logrotate-examples/>

3.7 Testing the OXSEED Transformation Server

In order to test the OXSEED Transformation Server, type in your favorite browser:

http://<server>:<port>

This should display the OXSEED Transformation Server Version as well as the version of the installed plugins.

If the output looks similar to the one below, the OXSEED Transformation Server is running and ready to process requests.

Welcome to OXSEED Transformation Server v2.0.2

For more information check the [OXSEED NodeJS Plugin Server User Guide and Reference](#)

[Core Modules](#) | [Plugin Modules](#) | [Deploy Service](#) | [Test Service](#)

Plugin Modules

Search:

Name <small>Name Ascending</small>	Version	Enable	Description	Log Level
oxsnps-afp2any	2.0.7	On	OXSEED NodeJS Server afp2any Plugin	INFO
oxsnps-msoffice2any	2.2.1	On	OXSEED NodeJS Server msoffice2any Plugin	INFO
oxsnps-pdf2pdf	2.2.1	On	OXSEED NodeJS Server pdf2pdf Plugin	INFO
oxsnps-segserver	2.0.9	On	OXSEED NodeJS Server segserver Plugin	INFO
oxsnps-soap	2.0.5	On	OXSEED NodeJS Server soap Plugin	INFO

Copyright © Maas Holding GmbH, Germany. All rights reserved.

3.8 Updating the OXSEED Transformation Server and Plugins

In order to update the OXSEED Transformation Server, type in your favorite browser:

<http://<server>:<port>/deploy>

This should display the following.

Welcome to OXSEED Transformation Server v2.0.2

For more information check the [OXSEED NodeJS Plugin Server User Guide and Reference](#)

[Core Modules](#) | [Plugin Modules](#) | [Deploy Service](#) | [Test Service](#)

Deploy Service

Select the File to deploy

No file chosen

Copyright © Maas Holding GmbH, Germany. All rights reserved.

- Click **Browse...** and select the OXSEED Transformation Server release file to deploy.
- Click **Upload & Deploy** to deploy.

The OXSEED Transformation Server will then deploy the selected release and finally restart.

4 List of public URLs

The table below describes all the OTS public URLs

URL	Description	Since Version
<server>:<port>/	displays the welcome to OTS page. It displays the OTS version as well as the version of its core modules and plugins	1.3.5u14
<server>:<port>/doc	displays the list of the available user guides	1.3.5u4
<server>:<port>/log	displays the content of the log directory	1.3.5u4
<server>:<port>/version	displays the welcome to OTS page. It displays the OTS version as well as the version of its core modules and plugins	1.3.4
<server>:<port>/history	displays OTS history.txt	1.3.5u17
<server>:<port>/transformservice	requests a synchronous transformation of either AFP2web supported documents, MS Office documents, secure PDFs or merge PDFs, using SOAP requests.	2.0.2
<server>:<port>/services/soap/transform	requests a synchronous transformation of either AFP2web supported documents, MS Office documents, secure PDFs or merge PDFs, using SOAP requests.	2.0.2
<server>:<port>/services/soap/asynctransform	requests an asynchronous transformation of AFP2web supported documents, using SOAP requests.	2.0.2
<server>:<port>/services/soap/status	returns the status of an asynchronous transformations of AFP2web supported documents, using SOAP requests.	2.0.2
<server>:<port>/backup	displays the list of the backup files found in the backups directory	1.3.5u16

List of Figures

No table of figures entries found.

List of Tables

Table 1 node.js Installation.....	11
Table 2 node.js Version	11
Table 3 npm Version	11
Table 4 Installation of the JDK 7	11
Table 5 Edit /etc/profile.....	12
Table 6 Installation of python 2.x.....	13
Table 7 OTS installation.....	13
Table 8 Install the additional node.js modules.....	14
Table 9 Install PM2 node.js module	14
Table 10 Installing wkhtmltox on Ubuntu	15
Table 11 npsServer Configuration.....	15
Table 12 oxsnps-afp2any.json	16
Table 13 oxsnps-html2any.json.....	18
Table 14 oxsnps-msoffice2any.json	19
Table 15 dcom.json.....	20
Table 16 log4j.xml	21
Table 17 oxsnps-pdfp2pdf.json	22
Table 18 log4j.xml.....	22
Table 19 oxsnps-soap.json	23
Table 20 Service WSDL.....	24
Table 21 AFP2Any Request.....	26
Table 22 AFP2Any Response	28
Table 23 Office2any Request.....	28
Table 24 Office2any Response	30
Table 25 pdf2securepdf Request.....	30
Table 26 pdf2securepdf Response	32
Table 27 MergePDF Request.....	32
Table 28 MergePDF Response	34
Table 29 Office2any Request.....	34
Table 30 Office2any Response	35
Table 31 Starting OTS	36
Table 32 Checking OTS Log	36
Table 33 OTS Logrotate Definition File	37

Table 34 Activating OTS Logrotate Definition File38

I n d e x

No index entries found.