

## Download and Install

### Option 1 - Install using Docker

Tornado is available in Docker and can be easily set up in just a few steps. If Docker is suitable, please refer to:

<https://github.com/Docmosis/tornado-docker>

Using Docker means you do not need to install all the components separately as defined below.

### Option 2 - Install Manually

The Tornado software and other resources for Tornado can be downloaded from the Docmosis web site under the Resources area:

<https://www.docmosis.com/resources/tornado.html>

You will also need to obtain a license key to allow the server to run. Free trial licenses are available on request or by signing up to a free trial under the Docmosis “Try” menu on the web site.

### Install LibreOffice

Docmosis works best with LibreOffice and the Long Term Support (LTS) edition is the most appropriate LibreOffice edition for production environments. LibreOffice is available here:

<https://www.libreoffice.org/download>

Docmosis works with either the 32- or 64-bit editions. You must make sure you choose the 32-bit or 64-bit version that matches your platform and the version of Java you will be using. They should all be 32-bit or all be 64-bit.

### Install Java

Install the latest version of Java if you do not already have it.

<https://java.com/en/download/manual.jsp>

Note: if you have installed a 32-bit version of LibreOffice you will need a 32-bit version of Java.

Tornado is compatible with Java 17 (min compatibility is Java 8).

## Starting and Connecting

The following commands assume the Tornado distribution is called `docmosisTornado.war`. This name may be slightly different so you should adjust the following commands to refer to the file you downloaded.

### To launch the server

Launching the server is easy:

```
java -jar docmosisTornado.war
```

The download also contains example scripts:

```
startTornado.bat  
startTornado.sh
```

that can launch Tornado for you. The scripts contain comments about some common settings you can apply.

By default, Tornado will run on port 8080 and will create a default configuration. The server will log what it is doing to the screen. Later when you have completed configuration, logging will move to the configured location.

## 32-Bit vs 64-Bit

If you see warnings about 32-bit or 64-bit issues, remember the golden rule:

You must make sure you choose the 32-bit or 64-bit version of LibreOffice that matches your platform and the version of Java you will be using. They should all be 32-bit or all be 64-bit.

On Windows, if LibreOffice is installed under “C:\Program Files” – then it is a 64-bit version and you will need to use a 64-bit Java installed in “C:\Program Files”. If LibreOffice is installed under “C:\Program Files (x86)” – then it is a 32-bit version and you will need to use a 32-bit Java installed in “C:\Program Files (x86)”.

On OSX, the Java installed is typically a 64-bit version so you should install a 64-bit LibreOffice.

## Configuring Tornado

Tornado provides multiple ways to set the configuration:

- a) Minimal command line interaction and manual configuration.
- b) Full configuration from command line parameters and environment settings.

To get up and started quickly to test Tornado, you probably will simply run tornado using a) above. The scripted/parameterised specifics ( b ) are described later in this document.

When you launch Tornado using the startTornado.bat or startTornado.sh script or the command line:

```
java -jar docmosisTornado.war
```

you should see some initialisation information then a console line like this:

```
INFO: Tornado started. Listening on 8080
```

At this point, the Tornado web console is ready, and further configuration can be done manually with a web browser. You can connect to Tornado using any web browser by visiting this address:

```
http://localhost:8080/
```

This will present you with the Tornado Server configuration page on your first visit:

**Configuration** | **Status**

*Mandatory Settings*

License Key \*  — Key Obtained from Docmosis

Open/Libre Office location \*

Source Templates From \*  Directory  AWS Bucket  Azure Blob Store

Working Area \*  — Working area for logs and caches

---

*Optional Settings*

Admin Password  — Optional Password to access this Console  
 show

Access Key  — Optional Access Key that clients must use to access the REST services of this Server  
 show

Template Markup Prefix

Template Markup Suffix

Custom Settings  — Custom Property settings for Docmosis

Enable Mail service  Enable Mail Server  Enable StartTLS  Enable SSL

Mail Server Host

Mail Server Port  — Mail Server Settings - if you wish to render documents directly to email

Mail Server User Name

Mail Server User Password

Mail Server Connection Timeout

Mail Server Email Sender Address

You will need to configure at least the fields with the asterisk (“\*”) marking them. Docmosis will pre-fill in examples to try to make it helpful.

If you are having trouble starting Tornado and getting even the initial start working, the following configuration parameters may help.

## Choosing a Different Port

If you have a requirement to use a different port for the server, you can override the default as follows:

```
java -Dport=8090 -jar docmosisTornado.war
```

## Enabling Debug Logging

To get more diagnostics in the start up sequence DEBUG logging can be enabled as follows:

```
java -Dlog.level=debug -jar docmosisTornado.war
```

## All Configuration Options

There are many other configuration options that can be. Many can be set manually in the Configuration tab of the Tornado console. All parameters can be set from the environment or the command line. The options are listed in detail at the end of this guide.

## Where Configuration Information Lives

Configuration information is saved against the account of the user running the server. This configuration is persisted in the Windows registry on Windows platforms and in the user's home directory on Linux/Unix based systems.

Windows:

*HKEY\_CURRENT\_USER\Software\JavaSoft\Prefs\com\docmosis\webserver*

Linux:

*~/java/.userPrefs/com/docmosis/webserver*

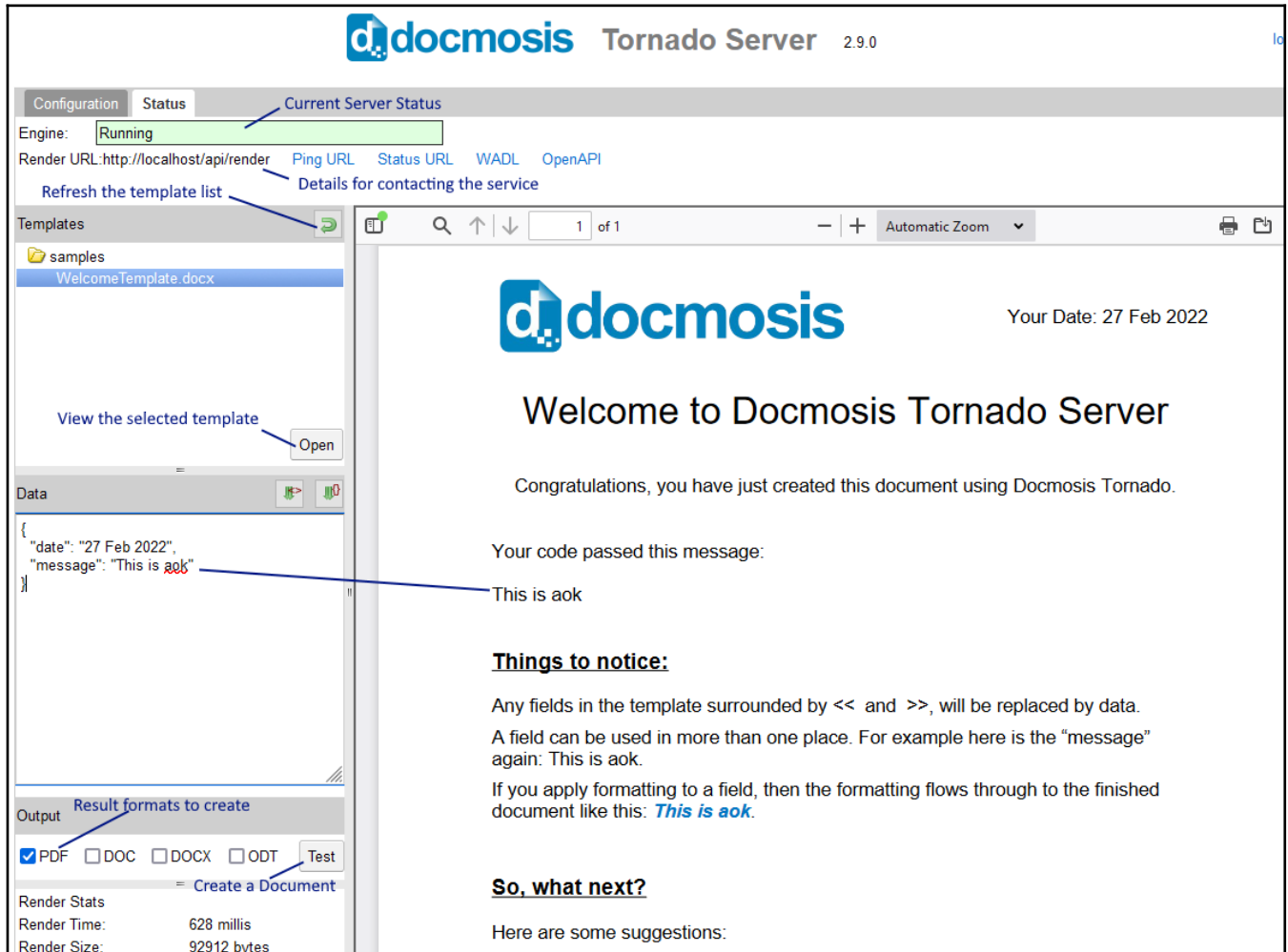
Mac

*/Library/Preferences/com.docmosis.webserver.plist*

## Running the Server and Testing

Once you have completed and saved the Configuration Page, you can move to the Status page, to start the server.

In future, the server will start automatically when you run the `java -jar docmosisTornado.war` command.



The screenshot displays the Docmosis Tornado Server web interface. At the top, the Docmosis logo and 'Tornado Server 2.9.0' are visible. The interface is divided into several sections:

- Configuration / Status:** Shows the engine status as 'Running'. It includes links for 'Render URL: http://localhost/api/render', 'Ping URL', 'Status URL', 'WADL', and 'OpenAPI'. A 'Refresh the template list' button is also present.
- Templates:** A list of templates is shown, with 'WelcomeTemplate.docx' selected. An 'Open' button is available for the selected template.
- Data:** A JSON object is displayed: 

```
{ "date": "27 Feb 2022", "message": "This is aok" }
```

. A blue arrow points from the 'aok' text in the JSON to the 'This is aok' text in the rendered document.
- Output:** A section for 'Result formats to create' with checkboxes for PDF (checked), DOC, DOCX, and ODT. A 'Test' button is also present.
- Render Stats:** Shows 'Render Time: 628 millis' and 'Render Size: 92912 bytes'.

The main content area shows the rendered document with the Docmosis logo, the date 'Your Date: 27 Feb 2022', and the heading 'Welcome to Docmosis Tornado Server'. Below the heading, it says 'Congratulations, you have just created this document using Docmosis Tornado.' and 'Your code passed this message:'. The rendered text 'This is aok' is shown in a standard font. A section titled 'Things to notice:' explains that fields in the template surrounded by << and >> will be replaced by data, and that a field can be used in more than one place. It provides an example: 'If you apply formatting to a field, then the formatting flows through to the finished document like this: *This is aok.*' A section titled 'So, what next?' follows with the text 'Here are some suggestions:'.

## Templates

The Tornado Status Page allows you to view a list of your templates and open them (if you are running the server on your local machine). You can then edit and save your templates so you can test the changes immediately. Note that this will change the templates in the path you have configured, so don't modify templates unless you mean to.

With AWS S3 and Azure Blob storage template locations, Tornado expects a “templates” and an “images” folder to be able to locate templates and images. If templates are not in a “templates” folder, no templates will be displayed.

## Creating Dummy Data

From the Tornado Status Page can create dummy data (either JSON or XML) based on the template you have selected. Tornado queries the template for fields and has its “best guess” at creating data that matches the template.

Templates can contain complicated structures for repeating and nested data, so you may need to adjust the generated data structure so that it looks like your expected data.

Tornado will generate data values: “value1”, “value2” and so on – which you can change to look more like your data.

## Creating a Document

You can create a document from the Tornado Status Page.

If you render a PDF only and your browser is configured with a PDF viewer, the output file will be displayed in the browser panel on the right.

If you choose any other formats, or combinations of formats, you will receive the rendered document as a download.

## Generating Documents from Your Application

The Tornado Status Page displays the Render URL to use for calling the Render service (just below the Engine status).

This is the URL to use with your client code / libraries to request documents to be rendered.

You should refer to the Tornado Web Services Guide in the Resources area of the Docmosis site <https://www.docmosis.com/resources/tornado.html> for details about invoking the render service.

## Comparing Tornado with Docmosis Cloud

Note that Docmosis Tornado provides only the render service. The Docmosis Cloud service provides other services to support producing documents in a cloud environment. The render service is identical to that provided by the cloud service except for:

1. The URL is different - you will direct the requests to your local Docmosis Tornado server instead of the public Cloud Service

2. The following REST web services are provided (more details are provided in the Tornado Web Services Guide):
  - a. “render” – create a document
  - b. “getTemplateStructure” – get a JSON description of the structure of a template
  - c. “convert” – convert the supplied document to another format (without any data merging)
  - d. “ping” and “status” – determine the Tornado operational status
3. Store-to directives for cloud and AWS (Amazon S3) storage are not available
4. REST clients do not need to supply an access key (unless you set one in the configuration tab)
5. Emailing documents is supported as long as you have configured an email gateway into Docmosis Tornado configuration

## Monitoring Tornado

Tornado includes two web service end-points to support automated monitoring:

“ping” e.g. <http://localhost:8080/api/ping>

“status” e.g. <http://localhost:8080/api/status>

See the Tornado Web Services Guide for details about these monitoring end-points.

## More Help

Docmosis document generation provides a large number of features controlled from both the templates and from the data. To get the most out of Docmosis, please read the Developer Guide and Template Guides on the Docmosis web site under the Resources area:

<https://www.docmosis.com/resources/all-resources.html>

We hope you enjoy using Tornado.



## Setting Configuration Options

Tornado configuration options can be provided in 3 ways (highest to lowest priority):

- a) The Java command line using the `-Dxxx=yyy` parameter format. Earlier examples show this style being used to enable debug logging or changing the port
- b) Environment variables of the format of `Docmosis_xxx=yyy`
- c) Previously Saved configuration settings which are stored using Java preferences and so vary from platform to platform.

Several options can be manually set using the Tornado web console configuration tab too and then saved. Note however, any changes may be overwritten at the next launch if parameters or environment variables supply overrides.

When Tornado is started, any configuration provided by the `-D` parameters will override any environment variables and both override any previously saved configuration. When an option is set using the environment or command line, it is applied and automatically saved. This means you don't need to provide the configuration settings every time.

For example, to set the templates folder, the following options can be used:

- a) Command line

```
java -DtemplatesDir=/projects/tornado/templates -jar docmosisTornado.war
```

- b) Environment variable

Linux/MacOs

```
export Docmosis_templatesDir=/projects/tornado/templates
```

```
java -jar docmosisTornado.war
```

Windows

```
set Docmosis_templatesDir=/projects/tornado/templates
```

```
java -jar docmosisTornado.war
```

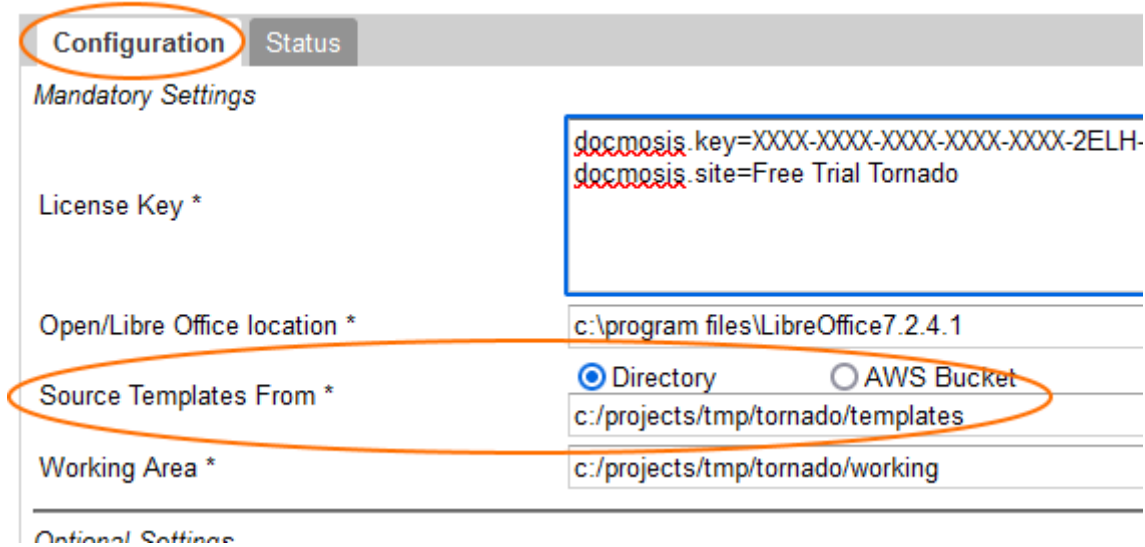
Environment variables may also be set in all upper case:

```
DOCMOSIS_TEMPLATESDIR
```

or with lowercase "docmosis\_" prefix:

```
docmosis_TemplatesDir
```

c) Tornado Web Console



## Runtime Help

You can get Tornado to dump the parameters list:

```
java -jar docmosisTornado.war help
```

and if you need to clear the configuration entirely for some reason:

```
java -jar docmosisTornado.war clearConfig
```

## All Configuration Options

### Common Settings

The following settings can be added to the Custom Settings on the Configuration page to enable

Setting	Description
port	Specify the port on which the console and the web services will listen
license	Specify the Tornado license all as one string. This includes the key and the site and overrides the key and site parameters below. “\n” is used to provide separate lines.  eg  license="docmosis.key=XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-X-XXXX\ ndocmosis.site=Free Trial Tornado"

key	<p>Specify the key part of the Tornado license. This requires the site or site1-3 parameters also.</p> <p>eg</p> <pre>key="docmosis.key=XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-X-XXXX" site="Docmosis.site=Free Trial Tornado"</pre> <p>or</p> <pre>key="docmosis.key=XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-X-XXXX" site1="Docmosis.site=Free Trial Tornado" site2="next line of site string" (if required) site3="another line of site string" (if required)</pre>
site	The full site string using "\n" to specify multiple lines as require (if key is multiple lines). Overrides the site1..site3 parameters and requires the key parameter.
site1	Specify the first line of the site
site2	Specify the second line of the site (if required)
site3	Specify the third line of the site (if required)
officeDir	Specify the office install location for LibreOffice
templatesDir	<p>Specify where templates will be sourced from (original templates)</p> <p>If specifying an AWS S3 Bucket or Azure Blob Store, Tornado will expect folders "templates" and "images" to be able to locate templates and images. If these folders are not present, no templates or images will be found.</p>
workingDir	Specify where logs and working caches are to be stored
adminPw	Specify the admin password for access the web console. Optional.
accessKey	Specify the access key for calling the web service end points. Optional.
customSettings	<p>Specify any custom settings using the format key=value and separating settings by "\n". For example:</p> <pre>customSettings="docmosis.xyz=abc\ndocmosis.xyz.2=def"</pre>
templatePrefix	Specify the template field prefix. Defaults to <<. Must be at least 2 chars.
templateSuffix	Specify the template field suffix. Defaults to >>. Must be at least 2 chars.
installSamples	Specify whether to install sample templates at startup. Defaults to true.

## Control of Logging

Logging of information by Tornado can be controlled by several command line settings:

Setting	Description
log.level=debug info error	<p>Specify the level of logging to the console and log files.</p> <p>eg: java -Dlog.level=debug -jar docmosisTornado.war</p>

<code>java.util.logging.config.file=path</code>	Specify the Java Util logging configuration file. Overrides log.level. eg: <code>java -Djava.util.logging.config.file=c:/logging.properties ..</code>
<code>log.dir.override</code>	Override the location where logs are to be written (default is <working area>/logs

## Enabling SSL/TLS Encryption

The following settings can be added to the Custom Settings on the Configuration page to enable SSL/TLS Encryption:

Setting	Description
<code>ssl.port=port</code>	The port to listen for secured connections.
<code>javax.net.ssl.keyStore=path</code>	The path to the key store file.
<code>javax.net.ssl.keyStorePassword=password</code>	The key store file password.
<code>javax.net.ssl.trustStore=path</code>	The path to the trust store file.
<code>javax.net.ssl.trustStorePassword=password</code>	The trust store file password.
<code>http.disable=true false</code>	Determines whether the non-secure listener should be disabled. Defaults to false.

Please see your Java documentation for more information on the specifics of what these settings do.

## Other Network Settings

The following settings can be added to the Custom Settings on the Configuration page to enable SSL/TLS Encryption:

Setting	Description
<code>address.listen</code>	Specify the address on which to listen for connections. This is useful when multiple networks are present and only a specified network should accept connections. By default, all networks are used.
<code>keepAlive</code>	Specify whether to set KeepAlive (SO_KEEPALIVE) on network connections. By default this is disabled. Values are true / false.

## Enabling Email from Tornado

Setting	Description
<code>mailEnabled</code>	Enable the mail server. Default is false.
<code>mailHost</code>	The mail server hostname
<code>mailPort</code>	The mail server port
<code>mailUser</code>	The mail server user name

<code>mailPw</code>	The mail server password
<code>mailFrom</code>	The from email address
<code>mailTimeout</code>	The mail server connect-timeout in milliseconds
<code>mailUseTls</code>	Enable TLS security on the connection to the mail server. Default false.
<code>mailUseSsl</code>	Enable SSL security on the connection to the mail server. Default false.
<code>mailConnectRetryMaxTimes</code>	Set the maximum number of attempts to connect to the mail server. Default 2.
<code>mailConnectRetryMinWaitMillis</code>	Set the minimum wait time to get a connection to the mail server. Default 5000ms.
<code>mailConnectRetryMaxWaitMillis</code>	Set the maximum wait time to get a connection to the mail server. Default 20000ms.
<code>mailConnectRetryRebuildTransport</code>	Rebuild the Message Transport object on failure to connect. Default true.
<code>mailSendRetryMaxTimes</code>	Set the maximum number of attempts to send email. Default 2.
<code>mailCustomHeadersAdd</code>	Add a custom mail header. Default true.
<code>mailCustomHeadersName</code>	Set the custom header name. Default X-DWS-Tag-1