

## Installing Adapt Authoring on Windows Server 2019

---

Author: Andy Cranston (andy@cranstonhub.com)

Revison: 0.1.0

Last edited on: 22nd September 2020

Assumptions / pre-requisites

---

Windows Server 2019 was installed with just the core environment.

The server should have a static IP address.

The server should be able to resolve domain names.

The server should have internet access.

Install git

---

From the Git for Windows website:

<https://git-scm.com/download/win>

download the 64 bit Git for Windows Portable EXE installer. At the time of writing it was version 2.28.0 and the downloaded file was called:

PortableGit-2.28.0-64-bit.7z.exe

Copy it to the server and put it in a directory.

TIP: I create a directory called "C:\opensrc" and put installers like this in that directory.

From a privileged command prompt (cmd.exe) change to the directory which has the installed and type:

PortableGit-2.28.0-64-bit.7z.exe

The suggested installation directory will be:

C:\opensrc\PortableGit

change this to:

C:\PortableGit

then click ok. The files are extracted.

Change to the following directory:

C:\PortableGit\bin

Run this command:

git --version

The output should be something like:

git version 2.28.0.windows.1

Do some basic git configuration:

```
git config --global user.name "someuser"
git config --global user.email "someuser@example.com"
```

Change "someuser" and "someuser@example.com" to match your setup.

Remember that you need to add:

```
C:\PortableGit\bin
```

to the PATH everytime you want to use it in a new command or Powershell window.

Install Node.js

-----  
From:

```
https://nodejs.org/en/download/
```

download the LTS line. Select the 64 bit Windows Installer (.msi) download. At the time of writing the version was 12.18.3 (including npm version 6.14.6). The downloaded file was:

```
node-v12.18.3-x64.msi
```

Copy this to a directory on the Windows server (for example "C:\opensrc").

Open a command prompt on the Windows server. Type:

```
cd \opensrc
node-v12.18.3-x64.msi
```

At the "Node.js Setup" window click "Next".

At the "End-User License Agreement" window tick the "I accept the terms in the License Agreement" check box and click "Next".

At the "Destination Folder" window the default should be "C:\Program Files\nodejs\". Click "Next".

At the "Custom Setup" screen click "Next".

At the "Tools for Native Modules" window leave the check box to automatically install the necessary tools unticked.

At the "Ready to install Node.js" window click "Install".

At the "Completed the Node.js Setup Wizard" click "Finish".

Close the command window.

Open a new command window.

Run:

```
echo %PATH%
```

Make sure the following two directories are in PATH and in this order:

```
C:\Program Files\nodejs
C:\Users\Administrator\AppData\Roaming\npm
```

Install the grunt CLI

-----

Type:

```
npm install grunt-cli -g
```

It is ok to ignore warnings like these:

```
npm WARN deprecated ....
```

At time of writing installing grunt-cli generated two of these warnings for:

```
urix@0.1.0  
resolve-url@0.2.1
```

Install MongoDB

From the:

```
www.mongodb.com
```

website download the Windows MSI installer for version 4.4 of the community edition of MongoDB. Copy this to a directory on the windows server (again put it in C:\opensrc).

The downloaded file should be something like:

```
mongodb-windows-x86_64-4.4.0-signed.msi
```

From a command prompt:

```
cd \opensrc  
mongodb-windows-x86_64-4.4.0-signed.msi
```

At the "Welcome to the MongoDB 4.4.0 2008R2Plus SSL (64bit) Setup Wizard" window click "Next".

At the "End-User License Agreement" window tick the "I accept the terms in the License Agreement" check box and click "Next".

At the "Choose Setup Type" click "Complete".

At the "Service Configuration" leave the defaults as they are and click "Next".

The defaults are:

```
Checkbox "Install MongoDoD as a Service" ticked.  
Radio button "Run Service as Network Service User" selected.  
Field "Service Name:" set to "MongoDB".  
Field "Data Directory:" set to "C:\Program Files\MongoDB\Server\4.4\data\".  
Field "Log Directory:" set to "C:\Program Files\MongoDB\Server\4.4\log\".
```

At the "Install MongoDB Compass" screen clear the "Install MongoDB Compass" check box. Click "Next".

At the "Ready to install MongoDB 4.4.0 2008R2Plus SSL (64 bit)" window click "Install".

At the "Completed the MongoDB 4.4.0 2008R2Plus SSL (64bit) Setup Wizard" window click "Finish".

Open the "Task Manager" and on the "Processes" tab look under the "Background processes" and there should be a process called:

```
MongoDB Database Server
```

Close the "Task Manager".

Install Adapt Authoring

-----  
From the command prompt:

```
PATH=%PATH%;C:\PortableGit\bin
cd \
git clone https://github.com/adaptlearning/adapt_authoring.git
```

A subdirectory called "adapt\_authoring" should have been created. Change into it:

```
cd adapt_authoring
```

Start the node package installation process:

```
npm install --production
```

This installs the Node package.

Now run the Adapt install.

```
node install
```

When prompted:

```
Continue? (Y/n)
```

respond:

```
Y
```

When prompted:

```
? Server port (5000)
```

press return.

When prompted:

```
? Server name (localhost)
```

respond with the hostname of the machine. For example:

```
win2019
```

When prompted:

```
? Data directory path (data)
```

press return.

When prompted:

```
? Git repository URL to be used for the authoring tool source code
(https://github.com/adaptlearning/adapt_authoring.git)
```

press return.

When prompted:

```
? Git repository URL to be used for the framework source code (https://github.com
```

/adaptlearning/adapt\_framework.git)

press return.

When prompted:

? Specific git revision to be used for the framework. Accepts any valid revision type (e.g. branch/tag/commit) (tags/v5.7.0)

press return.

When prompted:

? Master database name (adapt-tenant-master)

press return.

When prompted:

? Will you be using a full database connection URI? (all connection options in the URI) (y/N)

respond:

N

When prompted:

? Database host (localhost)

press return.

When prompted:

? Database server port (27017)

press return.

When prompted:

? Database server user (only specify if using database authentication) ()

press return.

When prompted:

? Database server password (only specify if using database authentication) [hidden]

press return.

When prompted:

? Database server authentication database (only specify if using database authentication) ()

press return.

When prompted:

? Will you be using an SMTP server? (used for sending emails) (y/N)

respond:

N

When prompted:

```
? Set q unique name for your tenant (master)
```

press return.

When prompted:

```
? Set the display name for your tenant (Master)
```

respond with the name of your company, your organisation name or anything that will look nice on the Adapt Authoring login page. For example:

```
ACME Incorporated
```

The framework is now cloned from githb.com - it can take a minute or two.

Next you are prompted for an email address to use as the "Super Admin" account. Do not use your own email address as you want to be a "normal" user when you login. Something like:

```
admin@localhost
```

should suffice.

Now choose a password. Make this a "strong" password and make sure you do not lose it!

The web application now gets built.

Run the server

```
-----
```

Once the web application is built the Adapt Authoring run:

```
node server
```

When the message:

```
Server started listening on port 5000
```

is displayed try access the application from a browser with a URL similar to:

```
http://win2019:5000/
```

Change win2019 to the IP address of the windows server of, if you have a resolveable hostname use that hostname.

If necessary, add a rule to the Windows firewall

```
-----
```

If there is no response it is most likely the firewall on the windows server. From a power shell prompt run:

```
Set-NetFirewallProfile -Profile Domain, Public, Private -Enabled False
```

and then try accessing the URL again. If this works then the issue is definitely the fire wall.

First of all enable the firewall again.

```
Set-NetFirewallProfile -Profile Domain, Public, Private -Enabled True
```

Now add and enable a rule with the display name "Adapt5000" with:

```
New-NetFirewallRule -Action Allow -Description "Inbound on port 5000 for Adapt"  
-Enabled True -Direction Inbound -LocalPort 5000 -Protocol TCP -DisplayName "Adapt5000"
```

Try the access again. Hopefully it should work.

Summary and next steps

-----

You should now have a working Adapt Authoring server running on a Windows 2019 server.

The next steps are to create users within Adapt and have them start creating content. Good luck!

-----

End of document.