
Numbas LTI provider Documentation

Release 2.1

Numbas

Feb 17, 2020

Contents

1	Contents	3
1.1	How it works	3
1.2	Getting started	3
1.3	Installation	4
1.4	For administrators	14
1.5	For instructors	19
1.6	For students	30
1.7	Glossary	31
	Index	33

This is a [Basic LTI 1.1](#) tool provider, to run [Numbas](#) exams in any LTI-compatible virtual learning environment.

The tool handles attempt data, and as well as offering CSV exports of student scores can report scores back to the host VLE's gradebook.

1.1 How it works

- *Install the software* on your own server, or on the cloud platform Heroku.
- *Create an LTI consumer key* and add the LTI tool to your VLE using the details provided.
- Add a link to the tool from your course's page on the VLE and upload a Numbas exam package.
- Students click on the link to attempt the exam. Their answers and scores are saved as soon as they are submitted. Students can leave the exam and resume it later by clicking on the link again.
- Instructors who click on the link can *review students' attempts, download a summary of scores, or report scores back to the VLE's gradebook.*

1.2 Getting started

This document will lead you through the process of setting up an instance of the Numbas LTI tool provider on your own server, and adding a Numbas activity to a course in your VLE.

1.2.1 What do I need in order to use this?

You must run your own instance of this tool - as well as using a fair amount of server resources, we don't want to keep other people's student data!

To install the tool, you need:

- A dedicated server to install the software on, which will communicate with your virtual learning environment. At Newcastle, we use servers running Ubuntu, though anything that can run Python will do. As a rough guide, you'll need around 0.5MB of disk space per student attempt.
- The ability to add a link to an LTI tool to your virtual learning environment. In Blackboard and Moodle, only administrators can do this.

At most institutions, this will require the help of your IT team.

If you're unsure whether you can use the LTI tool provider, or want help setting it up, [email the Numbas team](#).

1.2.2 Install the software

Follow the *installation instructions* to set up an instance of the Numbas LTI tool provider on your own server.

1.2.3 Link your VLE

Create a consumer in the tool provider's admin interface, and connect to it from your VLE.

This may require support from the administrator of your VLE.

1.2.4 Link a Numbas editor

Note: This step is optional.

You can *link to an instance of the Numbas editor* to make a selection of exams available directly from the LTI tool, instead of having to download a Numbas exam package to your own computer first.

1.2.5 Add a Numbas activity to your course

Add a link from your course's page in your VLE to the Numbas tool provider.

Download a Numbas exam package from [the Numbas editor](#) and upload it to the newly-created resource.

Students who click on the link will be shown the exam, and their answers and scores will be saved to the tool provider.

1.2.6 Get scores for the activity

You can *download a .CSV file* of the students' scores or *report their scores back to the VLE's gradebook*.

1.3 Installation

We have written installation instructions for a selection of common platforms.

The Numbas LTI tool provider can be installed on any platform that can run Python; if you're installing on a platform not listed below, follow the *Ubuntu installation instructions* and make changes as necessary. [Contact the Numbas team](#) if you need help.

1.3.1 Installation on Ubuntu

You will need:

- Administrator access to a server with at least 4GB of RAM, connected to the internet. A virtual machine is fine.
- A domain name pointed at the server you're going to use. A subdomain (e.g. `numbas-lti.youruniversity.edu`) is fine.

- An SSL certificate: LTI content must be served over HTTPS. These can be obtained easily and for free from [Let's Encrypt](#).

These instructions will take a fresh machine running Ubuntu 16.04 and set up the Numbas LTI tool provider to run through NGINX or Apache. On different operating systems or with different web servers, the process will be different. There are alternate instructions for *installation on RedHat Enterprise Linux 7*.

We will set up:

- The Numbas LTI provider Django app running inside a virtual Python environment, isolated from the system's Python environment.
- A PostgreSQL database for the LTI provider to use.
- `Supervisord` will ensure the app is always running.
- The [NGINX webserver](#) as a reverse proxy to serve the LTI provider to the outside world.

Set up the environment

First, install packages, set up users, and create the required paths (you can save this script as a file and run it as root - it doesn't need any user input):

```
#!/usr/bin/bash

# set up user group
adduser --disabled-password numbas_lti
adduser www-data numbas_lti

# install packages
apt update
apt install \
    git redis-server postgresql postgresql-server-dev-all \
    libxml2-dev libxslt1-dev python-dev lib32z1-dev python3-pip supervisor
pip3 install virtualenv

# get the numbas-lti-provider code
git clone https://github.com/numbas/numbas-lti-provider.git /srv/numbas-lti-provider
chown -R numbas_lti:numbas_lti /srv/numbas-lti-provider

# create media and static file directories
mkdir /srv/numbas-lti-media
mkdir /srv/numbas-lti-static
chown -R numbas_lti:numbas_lti /srv/numbas-lti-media
chown -R www-data:www-data /srv/numbas-lti-static
chmod -R 770 /srv/numbas-lti-*

# create the virtualenv for the python modules
virtualenv -p python3 /opt/numbas_lti_python
chown -R numbas_lti:numbas_lti /opt/numbas_lti_python
chmod -R 770 /opt/numbas_lti_python

# install python modules
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
pip install asgi_redis psycpg2
```

Next, create a database and set a password to access it (replace `$password` with your chosen password in the following script):

```
sudo -u postgres psql -c "CREATE USER numbas_lti WITH ENCRYPTED PASSWORD '$password'
↳CREATEDB;"
sudo -u postgres createdb -U numbas_lti numbas_lti -h localhost
```

Configuring the Numbas LTI provider

Run:

```
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
python first_setup.py
```

This script will ask a few questions, and configure the Numbas LTI provider accordingly. It will set up the database, and create an admin user account which you will use to manage the LTI provider through its web interface.

Once you've run this script, the last remaining steps are to start the app, and then set up a webserver to expose it to the outside world.

Configure supervisord

Supervisord ensures that the Numbas LTI provider app is always running.

Save the following as `/etc/supervisor/conf.d/numbas_lti.conf`:

```
[program:numbas_lti_daphne]
command=/opt/numbas_lti_python/bin/daphne numbasltiprovider.asgi:channel_layer --port
↳87%(process_num)02d --bind 0.0.0.0 -v 2
directory=/srv/numbas-lti-provider/
user=www-data
autostart=true
autorestart=true
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE=numbasltiprovider.settings
numprocs=4
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_daphne_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_daphne_stdout.log

[program:numbas_lti_workers]
command=/opt/numbas_lti_python/bin/python /srv/numbas-lti-provider/manage.py runworker
directory=/srv/numbas-lti-provider/
user=www-data
autostart=true
autorestart=true
redirect_stderr=True
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings"
numprocs=10
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_workers_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_workers_stdout.log

[group:numbas_lti]
```

(continues on next page)

(continued from previous page)

```
programs=numbas_lti_daphne,numbas_lti_workers
priority=999
```

Once you've set this up, run:

```
systemctl restart supervisor
```

Supervisord will start the Numbas LTI provider, and restart it automatically if it ever crashes.

Set up a webserver

We have instructions for two web servers: *NGINX* and *Apache*.

With NGINX

NGINX is a high performance webserver, ideal for use as a reverse proxy. It is the recommended option for the Numbas LTI provider.

Install NGINX:

```
apt install nginx
```

Overwrite `/etc/nginx/sites-available/default` with the following:

```
upstream backend_hosts {
    server 0.0.0.0:8700;
    server 0.0.0.0:8701;
    server 0.0.0.0:8702;
    server 0.0.0.0:8703;
}

server {
    listen 443;
    client_max_body_size 20M;

    ssl on;
    ssl_certificate /etc/ssl/numbas-lti.pem;
    ssl_certificate_key /etc/ssl/numbas-lti.key;

    error_page 502 /502.html;
    location = /502.html {
        root /srv/www/server-error;
    }

    location /static {
        alias /srv/numbas-lti-static;
    }

    location /media {
        alias /srv/numbas-lti-media;
    }

    location / {
        proxy_pass http://backend_hosts;
```

(continues on next page)

(continued from previous page)

```
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
proxy_buffering off;
proxy_redirect off;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Host $server_name;
proxy_set_header X-Scheme https;
proxy_set_header X-Forwarded-Proto https;
proxy_read_timeout 600s;
}
}
```

Set the `ssl_certificate` and `ssl_certificate_key` lines to the paths to your SSL certificate and key files. If you're using certbot, it will add those lines for you.

You should put something in `/srv/www/server-error/502.html`, to be shown when there's a server error. This can happen if the Numbas LTI provider isn't running, or otherwise fails to communicate with NGINX.

With Apache

Apache is a very commonly-used webserver. While it can be used as a reverse proxy for the Numbas LTI provider, it's not great at dealing with the many simultaneous connections that the LTI provider requires. In some circumstances, Apache might be your only option, so the instructions are provided as a reference.

Install required packages:

```
apt install apache2
a2enmod ssl proxy proxy_wstunnel proxy_http proxy_connect headers rewrite
```

Overwrite `/etc/apache2/sites-available/000-default.conf` with the following:

```
<VirtualHost *:443>
  SSLEngine on
  SSLProxyEngine on
  SSLCertificateFile /etc/apache2/ssl/certs/numbas_lti.crt
  SSLCertificateKeyFile /etc/apache2/ssl/private/numbas_lti.key

  ProxyPreserveHost On
  ProxyRequests Off
  ProxyPass /static !
  Alias "/static" "/srv/numbas-lti-static"
  ProxyPass "/websocket" "ws://0.0.0.0:8707/websocket"
  ProxyPassReverse "/websocket" "ws://0.0.0.0:8707/websocket"
  ProxyPass / http://0.0.0.0:8707/
  ProxyPassReverse / http://0.0.0.0:8707/

  RequestHeader set X-Scheme "https"
  RequestHeader set X-Forwarded-Proto "https"

  <Directory "/srv/numbas-lti-static">
    AllowOverride None
```

(continues on next page)

(continued from previous page)

```
Options FollowSymLinks
Require all granted
</Directory>

ErrorLog ${APACHE_LOG_DIR}/numbas_lti.error.log
CustomLog ${APACHE_LOG_DIR}/numbas_lti.access.log combined
</VirtualHost>
```

Set the `SSLCertificateFile` and `SSLCertificateKeyFile` lines to the paths to your SSL certificate and key files. If you're using certbot, it will add those lines for you.

Obtain an SSL certificate

An SSL certificate allows your server to communicate with browsers securely.

The easiest way of obtaining an SSL certificate is with [certbot](#), from the EFF. It's a command-line tool which automatically acquires certificates from [Let's Encrypt](#) for any domains you're serving. Follow the instructions on the certbot site, after setting up your web server, to obtain a certificate.

These certificates don't last very long, and need to be renewed. You can do this automatically by running `certbot renew` as a cron job; put the following in `/etc/cron.daily/renew-certbot`:

```
#!/bin/sh
certbot renew
```

Make sure that `/etc/cron.daily/renew-certbot` is executable by the root user.

If you have no other way of obtaining a certificate, you can [create a self-signed certificate](#) which will produce a security warning in web browsers.

Updating the software

You should keep the software up-to-date with any bugfixes or new features.

Run the following:

```
cd /srv/numbas-lti-provider
git pull origin master
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
python manage.py migrate
python manage.py collectstatic --noinput
supervisorctl restart numbas_lti:
```

Ready to use

Once you've got everything running, the LTI provider will be available to use, at the domain name you configured.

Open the site in a web browser and log in using the admin account credentials you set up earlier.

The next step is to add an LTI consumer key so that your VLE can connect to the LTI provider.

1.3.2 Installation on RedHat Enterprise Linux 7

You will need:

- Administrator access to a server with at least 4GB of RAM, connected to the internet. A virtual machine is fine.
- A domain name pointed at the server you're going to use. A subdomain (e.g. `numbas-lti.youruniversity.edu`) is fine.
- An SSL certificate: LTI content must be served over HTTPS. These can be obtained easily and for free from [Let's Encrypt](#).

These instructions will take a fresh machine using RHEL 7 and set up the Numbas LTI tool provider to run through NGINX. On different operating systems or with different web servers, the process will be different. There are alternate instructions for *installation on Ubuntu*.

We will set up:

- The Numbas LTI provider Django app running inside a virtual Python environment, isolated from the system's Python environment.
- A PostgreSQL database for the LTI provider to use.
- `Supervisord` will ensure the app is always running.
- The `NGINX` [webservice](#) as a reverse proxy to serve the LTI provider to the outside world.

Set up the environment

Create the `/etc/yum.repos.d/nginx.repo` with the following contents:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/rhel/7/$basearch/
gpgcheck=0
enabled=1
```

First, set up a user account and install packages, and create the required paths. Run the following commands as root:

```
#!/bin/bash
useradd numbas_lti
yum install https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/
↳pgdg-redhat96-9.6-3.noarch.rpm
yum install postgresql96-server \
    python34 python34-pip python34-devel \
    redis supervisor nginx

systemctl enable redis
systemctl start redis

# create media and static file directories
mkdir /srv/numbas-lti-media
mkdir /srv/numbas-lti-static
chown -R numbas_lti:numbas_lti /srv/numbas-lti-media
chown -R www-data:www-data /srv/numbas-lti-static
chmod -R 770 /srv/numbas-lti-*

# get the numbas-lti-provider code
git clone https://github.com/numbas/numbas-lti-provider.git /srv/numbas-lti-provider
```

(continues on next page)

(continued from previous page)

```

chown -R numbas_lti:numbas_lti /srv/numbas-lti-provider

# create the virtualenv for the python modules
virtualenv /opt/numbas_lti_python
chown -R numbas_lti:numbas_lti /opt/numbas_lti_python
chmod -R 770 /opt/numbas_lti_python

# install python modules
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
pip install asgi_redis psycpg2-binary

```

Change PostgreSQL to use password authentication: edit `/var/lib/psql/9.6/data/pg_hba.conf`, and change:

```
host    all             all             127.0.0.1/32    ident
```

to:

```
host    all             all             127.0.0.1/32    md5
```

Now restart PostgreSQL and create a database:

```

systemctl restart postgresql-9.6
sudo -u numbas_lti createdb numbas_lti

```

Configuring the Numbas LTI provider

Run:

```

cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
python first_setup.py

```

This script will ask a few questions, and configure the Numbas LTI provider accordingly. It will set up the database, and create an admin user account which you will use to manage the LTI provider through its web interface.

Once you've run this script, the last remaining steps are to start the app, and then set up a webserver to expose it to the outside world.

Configure supervisord

Supervisord ensures that the Numbas LTI provider app is always running.

Save the following as `/etc/supervisord.d/numbas_lti.ini`:

```

[program:numbas_lti_daphne]
command=/opt/numbas_lti_python/bin/daphne numbasltiprotider.asgi:channel_layer --port_
↪87%(process_num)02d --bind 0.0.0.0 -v 2
directory=/srv/numbas-lti-provider/
user=numbas_lti
autostart=true
autorestart=true

```

(continues on next page)

(continued from previous page)

```
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE=numbasltiprovider.settings
numprocs=4
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_daphne_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_daphne_stdout.log

[program:numbas_lti_workers]
command=/opt/numbas_lti_python/bin/python /srv/numbas-lti-provider/manage.py runworker
directory=/srv/numbas-lti-provider/
user=numbas_lti
autostart=true
autorestart=true
redirect_stderr=True
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings"
numprocs=10
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_workers_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_workers_stdout.log

[group:numbas_lti]
programs=numbas_lti_daphne,numbas_lti_workers
priority=999
```

Once you've set this up, run:

```
systemctl restart supervisor
```

Supervisord will start the Numbas LTI provider, and restart it automatically if it ever crashes.

Set up the NGINX webserver

NGINX is a high performance webserver, ideal for use as a reverse proxy. It is the recommended option for the Numbas LTI provider.

Add the *nginx* user to the *numbas_lti* group:

```
usermod -a -G numbas_lti nginx
```

Overwrite `/etc/nginx/conf.d/default.conf` with the following:

```
upstream backend_hosts {
    server 0.0.0.0:8700;
    server 0.0.0.0:8701;
    server 0.0.0.0:8702;
    server 0.0.0.0:8703;
}

server {
    listen 80;
    client_max_body_size 20M;

    error_page 502 /502.html;
    location = /502.html {
        root /srv/www/server-error;
```

(continues on next page)

(continued from previous page)

```

}

location /static {
    alias /srv/numbas-lti-static;
}

location /media {
    alias /srv/numbas-lti-media;
}

location / {
    proxy_pass http://backend_hosts;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_buffering off;
    proxy_redirect      off;
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Host $server_name;
    proxy_set_header    X-Scheme https;
    proxy_set_header    X-Forwarded-Proto https;
    proxy_read_timeout  600s;
}
}

```

Set the `ssl_certificate` and `ssl_certificate_key` lines to the paths to your SSL certificate and key files. If you're using `certbot`, it will add those lines for you.

You should put something in `/srv/www/server-error/502.html`, to be shown when there's a server error. This can happen if the Numbas LTI provider isn't running, or otherwise fails to communicate with NGINX.

Finally, open the firewall to allow web traffic:

```

setsebool -P httpd_can_network_connect 1
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
setenforce permissive
systemctl start nginx

```

Obtain an SSL certificate

An SSL certificate allows your server to communicate with browsers securely.

The easiest way of obtaining an SSL certificate is with `certbot`, from the EFF. It's a command-line tool which automatically acquires certificates from [Let's Encrypt](#) for any domains you're serving. Follow the instructions on the `certbot` site, after setting up your web server, to obtain a certificate.

These certificates don't last very long, and need to be renewed. You can do this automatically by running `certbot renew` as a cron job; put the following in `/etc/cron.daily/renew-certbot`:

```

#!/bin/sh
certbot renew

```

Make sure that `/etc/cron.daily/renew-certbot` is executable by the root user.

If you have no other way of obtaining a certificate, you can [create a self-signed certificate](#) which will produce a security warning in web browsers.

Updating the software

You should keep the software up-to-date with any bugfixes or new features.

Run the following:

```
cd /srv/numbas-lti-provider
git pull origin master
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
python manage.py migrate
python manage.py collectstatic --noinput
supervisorctl restart numbas_lti:
```

Ready to use

Once you've got everything running, the LTI provider will be available to use, at the domain name you configured.

Open the site in a web browser and log in using the admin account credentials you set up earlier.

The next step is to add an LTI consumer key so that your VLE can connect to the LTI provider.

1.3.3 Installation on Heroku

[Heroku](#) is a cloud computing platform. They provide a basic free service which is a good option if you want to test the Numbas LTI tool for compatibility with your learning environment.

To use Heroku, you must create an account. Click on the button below to create your own instance of the Numbas LTI tool provider on Heroku's free service.

You'll be prompted to create an account, if necessary, and then click *Deploy app*. Once the app is deployed, it's ready to use: click the *View* button to begin the web-based setup.

This service is unlikely to be adequate for use with large numbers of students. Heroku offers a paid service which can scale up to meet demand, but make sure to check its costs and terms carefully.

1.4 For administrators

1.4.1 Consumers

A *tool consumer* is an application which sends users to the Numbas LTI provider, and receives score data in return. In most cases, this will be a Virtual Learning Environment (VLE), such as Blackboard, Canvas or Moodle.

Only admin users on the LTI provider can create and manage consumers.

Adding a consumer

The first step is to add a consumer. Click the *Add a new LTI consumer* button.

You will be asked to give a unique `_key_` to identify the consumer, and optionally the URL of the consumer's home-page. The key can be any text, but should uniquely identify the consumer and its purpose.

Add a new LTI consumer

Instructors will use the consumer key you enter below, and the automatically-generated consumer secret, to add the Numbas LTI tool to their course.

Consumer key

The key should be human-readable, and uniquely identify this consumer.

Home URL of consumer

Fig. 1: The “Add a consumer” screen.

Once you have created a consumer, you are shown its *key* and *shared secret*. These are the pieces of information that the consumer will use to establish a secure connection with the Numbas LTI provider.

You must now set up the connection from the consumer's end. There are instructions for some of the most popular VLEs below.

Blackboard

Follow [Blackboard's instructions](#) - you'll need to add Numbas as a new LTI Tool provider, and then add a placement for it.

Blackboard doesn't support automatic configuration by URL, so you'll have to use the manual configuration settings.

For the tool provider:

- The **Provider domain** is your launch URL.
- The **Tool provider key** is your consumer key.
- The **Tool provider secret** is your shared secret.

For the placement:

- The **Label** is “Numbas” (or whatever you want it to be)
- The **Type** is “Content Type”, and select “Assessments” in the menu.

The configured LTI consumers are listed below. To access this tool, use the configuration URL below, and the corresponding consumer key and consumer secret.

Consumer key	Shared secret			
my university moodle	fa8b3khw0sn8rco5alvu	0 resources.	Manage	Visit

[+ Add a new LTI consumer](#)

Configuration information

[Configuration by URL](#) [Manual configuration](#)

Here's the automatic configuration URL:

```
http://localhost:8001/config.xml
```

[Admin site](#)

Fig. 2: The consumer management screen.

- Tick **Placement allows grading**.
- Save the icon from your “Manage consumers” page and upload it under the **Icon** field on the Blackboard form.

You can now add a Numbas exam to any folder in Blackboard by selecting “Numbas” from the *Assignments* menu.

Moodle

Moodle from version 3.0 onwards supports LTI configuration by URL. Earlier versions require you to enter the configuration settings manually.

If you have administrator privileges on Moodle, you can add the tool as an activity type, so you don't have to enter the settings for each exam.

Moodle 3.0+

- Go to *Site administration* → *Plugins* → *Activity modules* → *External tool* → *Manage tools*.
- Paste your **automatic configuration URL** into the box labelled *Tool URL*, and click *Add*.
- Enter your **consumer key** and **shared secret**, then click “*Save changes*”.
- Go back to the *Manage tools* page and edit the Numbas tool.
- For *Default launch container*, select “New window”.
- If you'd like Numbas to show up as an activity type, bypassing the “External tool” step, for *Tool configuration usage*, select “Show in activity chooser and as a preconfigured tool”.

Now you can add a Numbas exam in any course by clicking *Add an activity or resource* and selecting *External tool*, then select “Numbas” from the *Preconfigured tool* dropdown.

Versions of Moodle before 3.0

- Go to *Site administration* → *Plugins* → *Activity modules* → *External tool* → *Manage tools*.
- Click on *Configure a tool manually*.
- Set *Tool name* to “Numbas”.
- The *Tool base URL* is your launch URL.
- The *Consumer key* is your consumer key.
- The *Shared secret* is your shared secret.
- Under *Tool configuration usage*, select “Show in activity chooser and as a preconfigured tool”.

Now you can add a Numbas exam in any course by clicking *Add an activity or resource* and selecting *Numbas*.

Canvas LMS

- Go to your course’s *Settings* page, then click on *Apps* and *View App Configurations*.
- Click the button to add an App.
- For *Configuration type*, select “By URL”.
- The *Consumer key* is your consumer key.
- The *Shared secret* is your shared secret.
- The *Config URL* is your automatic configuration URL.

Now you can add a Numbas exam in your *Modules* page:

- Click the button to add an item.
- Select *External tool* and click on *Numbas*.
- Tick *Load in a new tab*.
- Click *Add item*.

Managing a consumer

Click on the *Manage* button to see more detailed information about a consumer.

The management view lists all contexts provided by this consumer. Contexts usually correspond to courses in a VLE - all *resources* launched from the same course will appear under the same context here. Click on a context’s name to see all resources in that context.

Time periods

Time periods allow you to group a consumer’s contexts by when they were active. You could use this to group contexts by academic year, for example.

Click on the *Manage time periods* button at the bottom of the consumer management page. Each time period has a name and start and end dates.

Each context is assigned to the latest time period containing the date on which the context's first resource was created. If a context falls into a gap between non-overlapping time periods, it will be shown separately.

Time periods are used solely to group contexts in the display. If you change or delete a time period, no data will be lost.

1.4.2 Editor links

Editor links form a connection between the Numbas LTI tool provider and an instance of the Numbas editor, allowing instructors to select exams from a list inside the LTI tool, rather than downloading an exam package to their own computer and then uploading it to the LTI tool.

The public Numbas editor hosted by mathcentre provides a wealth of ready-made exams on a variety of subjects.

Creating an editor link

From the *admin interface*, click on the *Editor links* button at the top of the screen.

Click on the *Connect to a Numbas editor* button. In the *Base URL of the editor* field, enter the address of the Numbas editor.

The address of the public editor hosted by mathcentre is:

```
https://numbas.mathcentre.ac.uk
```

Once you've created an editor link, you can select the *projects* from the editor that you want to use.

Only projects which are *marked visible to non-members* are available. This is because the editor link has no authentication mechanism, and linked projects are available to all *consumers*.

Once you have selected some projects, published exams from those projects will be available for instructors to pick on the *new Numbas activity* screen.

1.4.3 Stress tests

Use the stress test feature to see how the LTI provider copes with high demand. You can use this to identify potential problems before any sessions where you expect many students to be attempting a Numbas exam at the same time.

During a stress test, each device taking part will open multiple connections to the LTI provider and simulate sending attempt data.

Running a stress test

To begin a new stress test, click the *Start a stress test* button. You will be taken to the stress test control panel.

On each device taking part in the test, log in to the Numbas LTI provider's admin interface and navigate to the stress test's control panel.

You can simulate a single attempt by clicking the *Start an attempt* button. Each simulated attempt will open a connection from your device to the LTI provider, and send a new data element at the rate specified by the *Set new elements every __ seconds* field.

You can start multiple attempts at the same time with the first line of the form. Click the *now* button to start some attempts immediately, or enter a time - you could use this to coordinate several devices.

Caution: Many web browsers have a limit on the number of websocket connections they can hold open at once. At time of writing, Chrome is limited to 255 simultaneous connections, Firefox to 200, and Internet Explorer is limited to 6.

If you simulate more attempts than this, the extra attempts will use the AJAX fallback.

You can change the rate at which new elements are set, and the number of elements set when each attempt starts.

A Numbas exam normally sets around 100 elements when an attempt starts. This number grows slightly for exams with more questions and question parts.

When a student submits an answer, three or four elements are set. So, a “realistic” value for the *Set new elements every ___ seconds* field will be around $\frac{\text{total attempt time}}{5 \times (\text{number of parts})}$.

Interpreting the results of a stress test

Summary is shown above the list of all simulated attempts. This is only for the attempts being simulated on your device - because the point of the test is to identify bottlenecks in the network, there’s no reliable way of showing data about all devices taking part in the test. For each attempt, the current state of the connection, and the number of elements unsaved, are shown.

All attempts should have finished “waiting to start” within a minute or so of the test beginning.

Ideally, there should be zero attempts without a websocket connection. Attempts without a websocket connection will use the AJAX fallback, which saves data less regularly.

The most important statistic is the number of attempts with unsaved data. If this number remains above 0 for a long time, it means that the server has a backlog of data to save. In a real test, each device will keep resending data until it gets an acknowledgement that it has been saved, so data will eventually be recorded, if slowly.

If the connection is lost entirely - for example, if there’s a power cut, the student’s computer crashes, or the student closes the test before all data has been saved - any data which has not been acknowledged by the server will be saved to the browser’s local storage on the student’s machine. When the student reopens the test, any data in local storage will be sent.

If an attempt shows *AJAX is not working*, then a request to use the AJAX fallback has failed: usually because the connection timed out or was refused entirely. This is a sign that the LTI provider can not cope with the amount of traffic you’re simulating, and students would see errors or data might go missing in a real test.

1.5 For instructors

1.5.1 Resources

A *resource* is a single Numbas exam, which students access via the VLE.

Resources are automatically created when you launch a new Numbas activity from the VLE as an instructor. You must select a Numbas exam to use, and then any students who launch the same activity will be shown the exam.

Creating a new resource

Either upload an exam package that you have downloaded from the Numbas editor or, if any *editor links* have been created, select an exam from the list.

New Numbas activity

This resource doesn't have a Numbas exam package associated with it yet.
Please upload one now or select from the list of ready-to-use exams.

Upload an exam package

To obtain an exam package, find your exam in the Numbas editor and click "Download SCORM package".

[Click here to open the Numbas editor.](#)

Package

No file selected.

Select a ready-made exam

Search

In

Numerical reasoning - ratio and percentage

by Christian Lawson-Perfect

Based on section 3 of the maths-aid/mathcentre workbook on numerical reasoning.

Fig. 3: The first thing you see on creating a new Numbas resource.

When you [download an exam package](#) from the Numbas editor, you must use the *SCORM package* option.

Once you've selected an exam, you will be shown the dashboard for the resource.

Dashboard

Must-Pass Test 1 | Dashboard | Attempts | Settings | Test run

332 attempts by 308 students.

Report scores back to VLE

Marking

Discount question parts

You can remark individual attempts on the attempts page.

Student progress

Download scores as CSV

Search for a student:

Name	Score	Attempts	Access tokens
Ami Tenanowad	100%	1	0
Kovalu Etinu	88%	1	0
Fube Alica	92%	1	0
Efere Medocowa	76%	1	0

Fig. 4: The dashboard screen.

When you open a resource as an instructor, you are first shown the dashboard. This view offers a few actions to do with the resource, and the navigation bar at the top has links to other views.

Report scores back to VLE

The LTI provider can automatically report scores back to the VLE.

Click the *Report scores back to VLE* button to begin this process. This may take some time; you'll be shown either a success message or any errors encountered while reporting scores.

This isn't supported by every VLE.

Discount question parts

Discounting a question part removes it from the score calculations: any marks students have earned for that part are discounted.

You might want to do this if an error is found in a question.

To discount a part, click on the *Discount a question part* button on the dashboard.

You are shown a list of all the question parts in the exam. Click a *Discount this part* button next to a part to discount it. You can choose whether to remove the part from the total available for the exam, or to award everyone full marks for the part. These have different effects on the weighting of other parts in the exam - removing a part from the total will increase the weighting of other parts, while awarding full marks will increase everyone's total score. Think carefully about what you want to do.

If you discount a gapfill part, all of its gaps are discounted. If you discount an individual gap, the other gaps are unaffected.

Discount question parts

Question	Part	Gap	
1			
1	a		Remove from total
1	a	0	Discount this part
2			
2	a		Discount this part
2	a	0	Discount this part
3			
3	a		Discount this part

Fig. 5: Discounting a question part.

Download scores as CSV

Click this button to download a `.csv` file containing the scores for each student who has attempt the resource.

The columns of the file are:

- First name
- Last name
- Email address
- Username
- Percentage score

The values in the name, email address and username fields come from the VLE. The username field might not correspond exactly to the student's username on the VLE; in particular, Blackboard prepends usernames with `cuid:`.

Student progress

From the dashboard, click *View individual student progress and grant access tokens* to view the *Student progress* table.

The *Student progress* table lists the names of students who have attempted the activity, along with their scores as calculated following the *grading method*, and the number of attempts they have made.

You can narrow down the displayed list by entering a name in the *Search for a student* box.

Note that only students who have launched the activity are listed - the LTI provider has no way of knowing about students who have access to the activity through the VLE but have never launched it.

Access tokens

When the number of attempts students are allowed to make is limited, circumstances can arise in which you want to allow particular students another attempt.

To do this, click the plus symbol in the *Access tokens* column next to the student's name in the *Student progress* table.

To revoke an access token, click the minus symbol next to the student's name in the *Access tokens* column. Attempts started with an access token that is subsequently removed will not be deleted.

When the student launches the activity, they will be offered the opportunity to start a new attempt.

Validate a receipt code

If enabled (see *Email attempt receipts to students on completion?*), students are emailed a receipt on completion of an attempt. The receipt contains information about their attempt, such as time and score, as well as a code that you can use to confirm that a receipt is valid.

On the *Validate a receipt code* page, copy a code from a student into a box.

If the code is valid, you'll be shown the information from the receipt, as well as a link to view the attempt's data in more detail.

If the code is invalid, or corresponds to an attempt at a different resource, you'll be told so.

Attempts

Click on the *Attempts* button at the top of the page to view the attempt management screen.

Download attempts summary as CSV

Click the *Download attempts summary as CSV* button to obtain a `.csv` file with information on every attempt at this activity.

The columns of the file are:

- First name
- Last name
- Email address

Must-Pass Test 1 [Dashboard](#) **Attempts** [Settings](#) [Test run](#)

Attempts

[Download attempts summary as CSV](#)

Search for a student:

Showing attempts 1 - 20 of 332 [Earlier »](#)

Student	Start time					Completion status	Score
Obefi Isufewawuwu	Jan. 11, 2018, 1:16 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	21.0 / 25.0 (84%)
Dat Dewok	Nov. 20, 2017, 1:03 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	21.0 / 25.0 (84%)
Iget Lumisora	Nov. 20, 2017, 12:07 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	17.0 / 25.0 (68%)
Ubobifosawu Kurofal	Nov. 20, 2017, 12:05 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	23.0 / 25.0 (92%)
Ogavehu Oraco	Nov. 20, 2017, 12:03 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	23.0 / 25.0 (92%)
Ledohu Orulec	Nov. 20, 2017, 12:01 p.m.	Review	Remark	SCORM data	Delete	completed (reopen)	25.0 / 25.0 (100%)

Fig. 8: The attempt management screen.

- Username
- Start time, in YYYY-MM-DD HH:MM:SS.ffffff+HH:MM format.
- Completed? (Either completed or incomplete)
- Total score
- Percentage (total score as a percentage of marks available)
- One column giving the total score for each question

Download all attempt data as JSON

Click the *Download all attempt data as JSON* button to obtain a .json file containing all data to do with attempts on this resource, including part and question scores, all SCORM data, completion status, and the values of question variables.

Review an attempt

Click the *Review* button to view a student’s attempt as they saw it.

This is useful when a student queries the mark they were awarded for a part of the exam.

Note that review mode always opens attempts as if they were completed, even if the student has not yet ended the exam.

Remark an attempt

Click the *Remark* button to manually change the score awarded for a question part.

You are shown a list of every question part in the exam. Click the pencil icon on the row corresponding to the part you want to change, and enter the new score. The new score is saved as you type, and the totals for the question and the whole exam are recalculated automatically.

Data

By clicking on the *Data* button, you can see a timeline describing actions the student took during their attempt.

Items shown include moving between questions, submitting answers, and marks awarded. This page will update in real-time as the student continues their attempt.

Click the *Download attempt data in JSON format* button to obtain a `.json` file containing all data pertaining to this attempt, including question and part scores, completion status, and the values of question variables.

Numbas uses the **SCORM** standard to store data about attempts. By clicking on the *View raw SCORM data for this attempt* button, you can see all of the SCORM data model elements stored for a particular attempt.

This is most useful for debugging connection errors, to confirm that data has been saved.

If *Most recent value only* is ticked, only the most recent value for each element is shown. Untick it to see every value that the element has taken since the start of the attempt.

You can type a regular expression in the *Search for an element* box to narrow down the displayed list of elements.

Delete an attempt

Click the *Delete* button to delete an attempt. This is permanent; the student will be able to start a new attempt next time they launch the activity.

Reopen an attempt

Sometimes students accidentally close their attempts before they mean to. Click the *Reopen* button to allow a student to complete their attempt. The next time that they launch the activity, they will be able to resume the attempt as if they had only paused it.

Beware that the standard Numbas settings allow a student to see the correct answers to every question once they have finished their attempt. If you're concerned about this, it's often better to make the student start a new attempt, rather than reopen the previous one.

Settings

Replace exam package

If you discover an error in your exam, you can update it by downloading it again from the editor and clicking the *Replace exam package* button.

Any new attempts will use the latest version of the exam package. Because the new version might have changed in a way that is incompatible with existing attempts, for example by removing or rearranging question parts, any attempts started with the old package will continue to use the old package.

Attempt by Nuhame Capege

Started 2018-05-03 10:18:52

Remark question parts

Question	Part	Gap	Score
1			
1	a		
1	a	0	<i>Discounted</i>
2			
2	a		
2	a	0	 <input type="text" value="3"/>  / 3.0
3			
3	a		
3	a	0	 0.0 / 3.0

Fig. 9: Remarking an attempt.

Settings

 Replace exam package

 Download exam package

Grading method

Highest score

Include incomplete attempts in grading?

Maximum attempts per user

0

When to show scores to students

Always

When to report scores back

Immediately

Save

Cancel

Fig. 10: The resource settings screen.

Grading method

Specify how a student's score for the activity is calculated.

- “Highest score” will use the highest total score from any of the student's attempts.
- “Last attempt” will use the total score from the attempt which the student began last.

Include incomplete attempts in grading?

If ticked, incomplete attempts will be included when calculating the student's score for the activity.

It's normally good to leave this on, so that students who forget to click the *End Exam* button won't be penalised.

Maximum attempts per user

How many *attempts* at the resource can each user take?

If set to 0, then there is no limit.

You can grant extra attempts to individual students with *access tokens*.

When to show scores to students

When a student reopens an activity, they are shown a summary of their attempts. You might not want to immediately show students their scores on this screen.

- “Always” means the student will see scores for all attempts, including incomplete attempts.
- “When attempt is complete” means the student will only see their score for an attempt once it is complete.
- “Never” means that no scores are shown to the student, even after they've completed their attempt.

Warning: This only controls the display of scores by the LTI provider. If you want to hide scores from the students, you must also turn off the [score feedback options in the exam editor](#).

When to report scores back

Specify when students' scores are reported back to the *consumer*. Some VLEs make reported scores available to students immediately, which you may not want.

- “Immediately” - scores are reported as soon as they change, i.e. whenever a student submits an answer.
- “On completion” - a student's score is reported when they complete an attempt.
- “Manually, by instructor” - Scores are only reported when an instructor clicks the *Report scores back to VLE* button on the dashboard.

Allow students to review attempts from

Specify when students are allowed to re-enter completed attempts in review mode.

If left blank, students can review their attempts at any time.

If a date and time are set, students may only review their attempts after that time.

Instructors may always review students' attempts, from the *Attempts* tab.

Email attempt receipts to students on completion?

If ticked, then when a student completes an attempt at this resource they will be emailed a receipt summarising their attempt.

The receipt contains a code which instructors can use to confirm the receipt's contents. See *Validate a receipt code*.

Test run

Click the *Test run* button to launch the Numbas exam. Data will not be saved - this feature is solely a convenience for instructors to check the contents of the exam.

1.6 For students

When a student clicks on a link to launch a Numbas *activity* for the first time, they are taken straight into the exam.

Their answers and scores are saved as soon as they are submitted.

Warning: Unsubmitted answers are not saved, so if the student's computer crashes, there will be no record of unsubmitted answers. It's important to make this clear to students during high-stakes tests, because they may want to wait until they've finished the whole question or exam before submitting. Numbas allows students to submit answers as many times as they like, with no penalty, so encourage them to submit as soon as they've entered an answer.

If the student leaves the test before completing it, they will be able to resume from where they left off next time they return.

On subsequent launches of the activity, the student is shown a summary of their attempts. If *the exam's settings* allow it, the score earned for each attempt is shown.

The student can click the *Continue* button to resume an incomplete attempt where they left off.

Once a student clicks the *End exam* button, the attempt is marked as "completed", and the student can not change their answers. If the server is configured to do it, the student is emailed a receipt containing information about their attempt, and a code that can be used by the instructor to *validate it*.

For completed attempts, the student can click *Review this attempt* to reopen the attempt in review mode, the same as was shown when they ended the exam. They will see the results summary screen, and can click on individual questions to see their answers and any feedback.

If *the exam's settings* allow it, or the student has an unused *access token*, they can start a new attempt by clicking the *Start a new attempt* button.

Must-Pass Test 2

Hi Nuham Capege. You have already attempted this activity.
You can resume or review your previous attempts, or start a new one.

Start time	Score	Status
May 3, 2018, 2:16 p.m.	0.0 / 42.0 (0%)	▶ Continue
May 3, 2018, 10:18 a.m.	3.0 / 42.0 (7%)	🍏 Review this attempt

[+ Start a new attempt](#)

Fig. 11: The attempt summary screen shown to a student.

1.7 Glossary

LTI Learning Tools Interoperability is a standard which defines how a *tool consumer* connects to a *tool providing* a learning activity. The consumer can provide some information about the person who launched the activity, such as their name, their role in the course, and a unique identifier. The *tool provider* can report back a score after a user has completed the activity.

VLE A Virtual Learning Environment, such as Blackboard, Moodle or Canvas.

Tool consumer A system from which users launch LTI activities, such as a *VLE*.

Tool provider A system which provides activities to *tool consumers*.

Resource In the Numbas LTI provider, each link from a tool consumer to the provider creates a new resource, representing a single exam to be completed by students.

Attempt A student may make one or more attempts at a *resource*. Each attempt is independent from the others.

Editor link A connection between the LTI provider and an instance of the Numbas editor.

A

Attempt, **31**

E

Editor link, **31**

L

LTI, **31**

R

Resource, **31**

T

Tool consumer, **31**

Tool provider, **31**

V

VLE, **31**