
GLaDOS

Release 1.0.12

Roman Gruber

Dec 02, 2022

HOWTOS AND MANUALS

1	Features	3
1.1	Exams	3
1.2	Tickets	10
1.3	Taking an Exam	14
1.4	Results	16
1.5	Example exam: Essay	21
1.6	Client crashes during exam	29
1.7	Restore a specific file	29
1.8	Hardware Recommendations	30
1.9	Installation Guide (From the sources)	31
1.10	Installation Guide (Debian package)	37
1.11	Softwareupdate on Debian	38
1.12	Upgrade from Debian 8 to 9	39
1.13	Upgrade from Debian 9 to 10	41
1.14	Glados config files	45
1.15	Exam client configuration	45
1.16	Network configuration	46
1.17	GLaDOS System Settings	47
1.18	Large exams with 200+ clients	47
1.19	LDAP Authentication	49
1.20	Active Directory Authentication (Simple)	52
1.21	Active Directory Authentication (Advanced)	52
1.22	LDAP with SSL	54
1.23	Test Login	56
1.24	User Migration	56
1.25	Multiple LDAP Servers and/or Active Directories	57
1.26	Placeholders	59
1.27	Login Scheme	59

GLaDOS is a fully configurable webinterface to take, manage and create exams using the [Lernstick](#).
The Project is available on [GitHub](#).

FEATURES

- Create exams in a *simple manner*
- Create *complex exams* with specific system configuration, additional software or permissions
- Manage your exams
- Screen capturing of the students screen
- Monitor exams in a *live view*
- Configure backup intervals for exams
- *Restore specific files* from the backup history during exams
- Generate conveniently *exam results* from the backups as a zip file
- *Submit* corrected exams back to the student

1.1 Exams


1.1.1 Create an exam

To start the exam creation, navigate to Actions->Create Exam. In the appearing wizard, you have to provide some information about the exam.

The *Name* and *Subject* fields are to identify the exam. The *Name* may not be unique, but it is not recommended to create multiple exams with the same name for reasons of clarity.

You can set a *Time Limit* (in minutes). Notice that this can also be set in the *ticket* and the value in the ticket will override this one.

For more information about the *Remote Backup Path*, please visit [this page](#).

There are multiple settings in the *Settings* tab. You can add new settings by clicking the *Add Setting* button. Choose the setting you want to add from the appearing list by clicking on it. You will now have the possibility to adjust (various) subsettings for the chosen setting. For more informations about the settings, please refer to the information given by clicking the  questionmarks aside.

For the *Screen capturing* setting, please read [Screen Capturing](#).

When clicking *Next Step*, a new field called *Exam File* will appear. Here you have to provide an image file holding the information for the exam. Please read the following sections on how to create this file:

- [Create a zip-file as exam file](#)
- [Create a squashfs-filesystem as exam file](#)

1.1.2 Create a zip-file as exam file

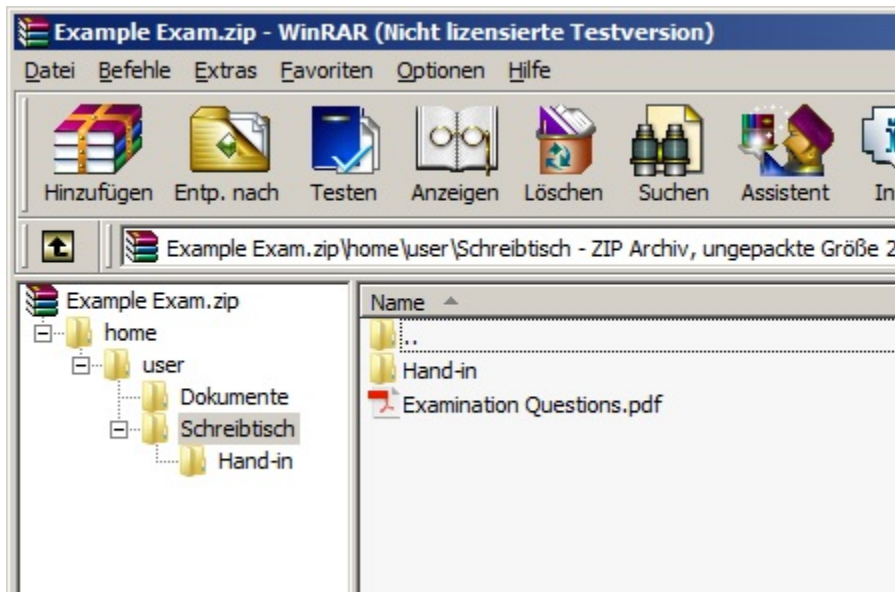
If your exam has a very simple structure, you can create a zip-file as exam template. For example, if you only need to put a file with your examination questions on the desktop and maybe some other resources somewhere else. If your exam does not need special configurations, installed/uninstalled applications or more complex system settings, then the zip-file is the easiest way to go.

If you decide to create a zip-file as exam template, make sure it holds the correct directory structure, so that your files are visible and accessible in the exam. All files in the zip-file will be extracted to the exam machine in that directory, where they were in the zip-file. If the directory does not exist, it will be created with default permissions. With a zip-file, it is not possible to create special files or set permissions on files (use a [squashfs-filesystem](#) for that).

The list below contains an example directory structure for an exam:

```
/home/  
/home/user/  
/home/user/Schreibtisch/  
/home/user/Schreibtisch/Examination Questions.pdf  
/home/user/Schreibtisch/Hand-in/  
/home/user/Dokumente/  
/home/user/Dokumente/Resource.pdf
```

The corresponding zip-file would look like this:

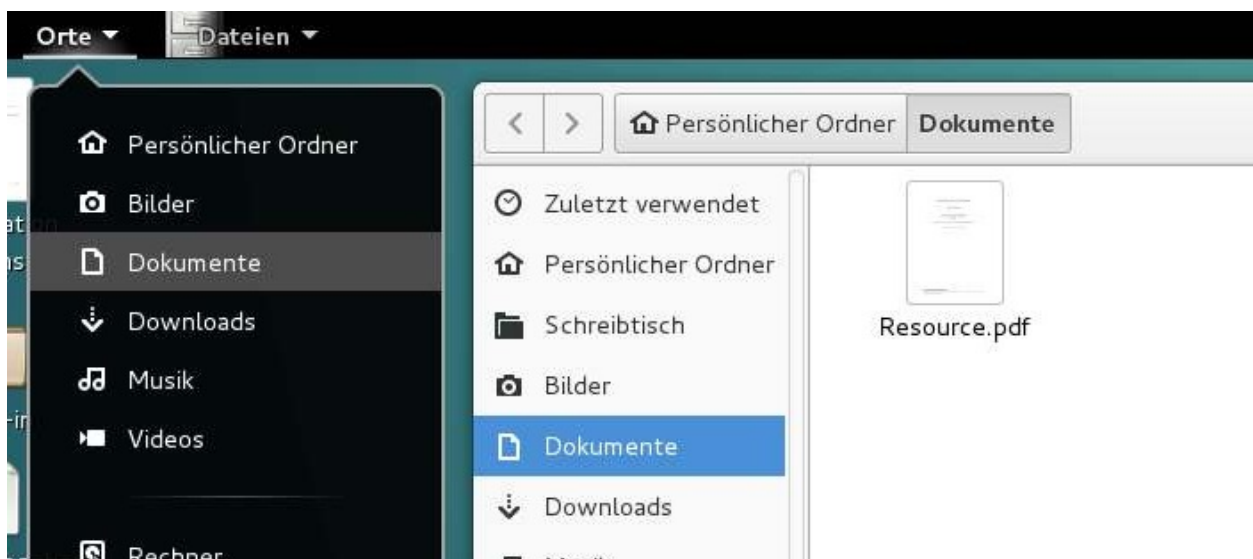


Example zip-file

If you want to put a file - say Examination Questions.pdf - at the Desktop, then drop the file in your zip file to the location /home/user/Schreibtisch/ as illustrated in the picture above. The testee will see all files in his/her exam under the directory, where you dropped them in the zip-file. The above zip-file will look in an exam as the pictures below.



Example Desktop 1



Example

Desktop 2

It is highly recommended to create a directory like Hand-in at the Desktop and advice the testee to put all results in this directory. Even though it might not be necessary, it will be easier to [collect the results](#) after the exam is done.

That's it! You can now upload the zip-file in the **Actions->Create Exam** wizard. If need a more complex configuration, then please have a look at [Create a squashfs-filesystem as exam file](#).

1.1.3 Create a squashfs-filesystem as exam file

You can also create an exam using a squashfs-filesystem. This is useful when your exam has a more complex form than just a few files at some place in the system. All kinds of exam configurations are possible with a squashfs-filesystem.

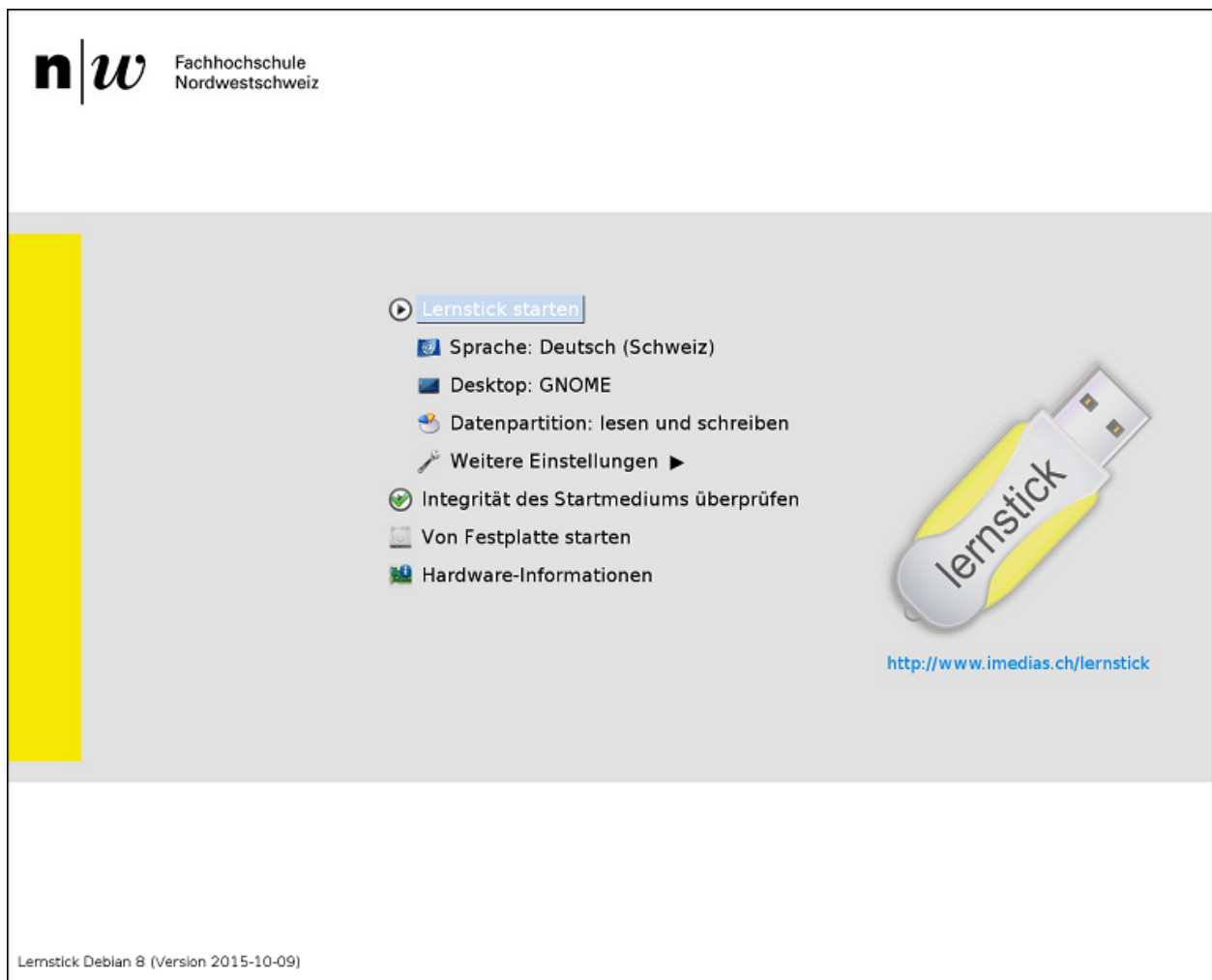
To create such an image file, please follow the steps.

Step 1

The first step is to boot [Lernstick Prüfungsumgebung](#) on your computer from USB. See the following instructions on how to boot your device from USB:

- [Start from USB-Device \(Mac\)](#)
- [Start from USB-Device \(Windows 10\)](#)

Please make sure to choose Datenpartition: lesen und schreiben, before starting the system:



Bootscreen

Step 2

Once the system has started, you can configure your exam. For example:

- install/uninstall specific applications
- preconfigure applications
- carry specific system configurations, which are not covered in the settings of the Actions->Create Exam wizard (Notice that, settings which are covered in the wizard will override settings you configure in the squashfs-filesystem).
- grant/deny advanced permissions to files and directories
- copy files needed in the exam to their locations
- ...

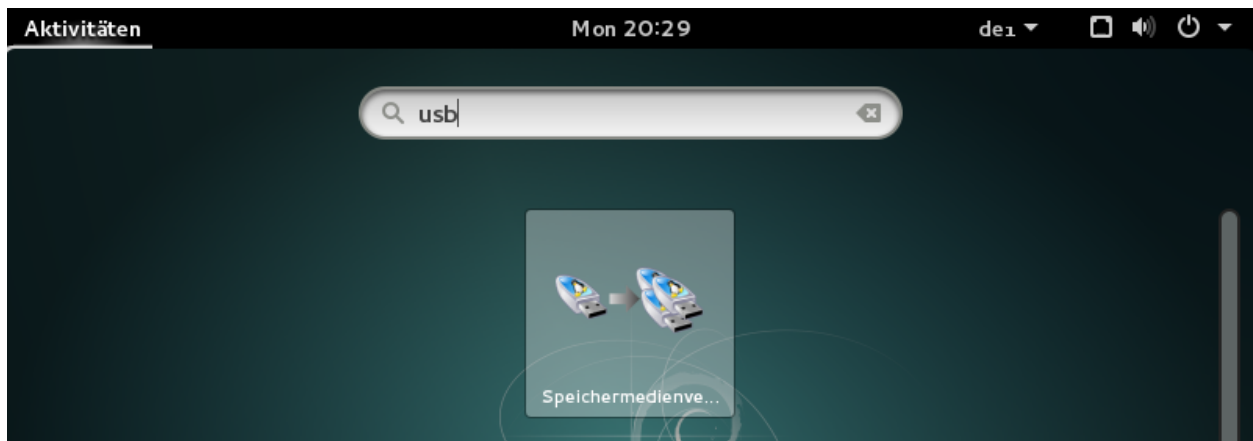
Step 3

When you finished with the setup for your exam, restart your computer.

This time it is important to choose Datenpartition: `nur lesen` from the start screen.

Step 4

Once the system has started again, open the application Speichermedienverwaltung.



Choose now `Das System in ein DVD-Abbild konvertieren`.



Check the box `nur Datenpartition`. You can leave the other options as they are. Press `Weiter` and the squashfs-filesystem will be generated.

Once the process has finished, the `.squashfs` file is then in `/media/Lernstick-Squashfs/lernstick.squashfs`

You can also start your own operating system and grab the file from the Exchange partition.

This is the file, which holds all information you configured in step 2. You can now upload it in the `Actions->Create Exam wizard`.

1.1.4 Remote Backup Path

This field in the `Create Exam wizard` is very important. It specifies the directory of the students machine to backup.

The default value is `/home/user` which points to the directory where all user related data is stored. In most cases the default setting will be sufficient.

Though, it is possible to change the value. If - for example - system log files need to be included in the backup (those are stored in `/var/log`), the default value `/home/user` will not be sufficient, since the users home directory does not contain `var/log`. In this case *Remote Backup Path* should be set to `/`. Notice that, **every** single file on the system would then be backed up regularly. This results in very huge backups and a lot of network traffic, compared to the default path. Therefore only do this if you really need to.

It is also possible to restrict the path even further than `/home/user`. Let's say, we only want to backup the files and directories on the Desktop. *Remote Backup Path* would then be set to `/home/user/Desktop`. Backups will be much smaller then, because a lot of files and directories in the users home directory would not be backed up anymore. But notice, if you do that, the *Screenshots* would not be backed up as well, because they are saved under `/home/user/.Screenshots`, which is not included in `/home/user/Desktop`. Also Libreoffice AutoRecovery information and backup copy would not be backed up (stored in `/home/user/.config`), therefore those options are absurd, when *Remote Backup Path* does not contain them.

Notice that the directory specified in *Remote Backup Path* must exist on the target system, else the backup process will fail.


1.1.5 The exam view

All data related to the exam can be observed in this view.

General

This tab shows general information about the exam, like the name and the subject. If you provided an exam file (See [Create a zip-file as exam file](#) and [Create a squashfs-filesystem as exam file](#)) you will see brief information about this file such as its size and type.

Settings

This is a list of exam specific settings you configured in the **Actions->Create Exam** wizard. For more information on each individual configuration option, please click the  Questionmark aside.

Please notice, all these settings will override the settings configured in the exam file!

Monitor

This tab can be useful during the exam, if you want to keep an eye on multiple tickets simultaneously.

Chart

This is a simple pie chart on the distribution of ticket states.

1.1.6 Monitoring exams

This describes the monitor exams view. It can be reached via **Actions->Monitor Exams** in the top navigation or if you are in the exam view, just click the **Monitor** tab.

Notice that, only tickets in **running** state will be shown in this view. The view is updating automatically if new tickets arrive at or leave the **running** state. Initially there might be no ticket running. The maximum number of screens visible on one page is 12.

By clicking on the name or token you will be redirected to the corresponding [ticket view](#). Besides the name or token, there is a hint on which window is currently focused/active on the students screen. You will see the icon as well as the name of the window as it is given by the running application (see example image below). If no window is active, there will be neither an icon nor a window name.

The red dot indicates that the system receives images live, whereas a gray dot indicates that the last received image is not live anymore.

Example Screen

This information as well as the image will update in a given interval. You can also click on the image itself. This will open a modal with a live image of the students screen in its native resolution. This image although will not update itself.

The update interval is specified by the [global setting](#) **Monitor refresh interval**. This can be changed in **System->Settings** if you have the permission.

1.2 Tickets

1.2.1 Create a single ticket

A ticket is the access authorization for the student to his exam. If you have 20 students to take the exam, you have to generate 20 tickets (See [Create multiple tickets](#)). You can create a single ticket or multiple at once for a given exam. Create a single ticket by the Actions->Create single ticket wizard and multiple tickets by the Actions->Create multiple tickets wizard.

The ticket *Token* (used to identify the exam, see [Taking an Exam](#)) is automatically generated. You can change it to a value of your desire, but notice that this value must be unique among all other ticket tokens (otherwise an error will occur).

Backup Interval describes the value (in seconds) for the backup schedule. It's the interval after which backup processes will run again on the ticket. 5 minutes is a moderate value for this.

Notice, this will increase network traffic, if set to a very low value.

You can set a *Time Limit* (in minutes) for your exam, but this will have no indication (nothing will happen though, if the time is up). In the [Ticket-view](#) can be seen whether the ticket is valid or not (time has expired).

This will override the setting [configured in the exam](#).

Each ticket can be assigned to a student in the *Test Taker* field.

Create the ticket by pressing Create at the bottom of the page. You will be redirected to the view page of the created ticket (See [The ticket view](#)).

Under Actions->Generate PDF you can generate a printable PDF file for this ticket. See the image below for an example ticket.


Ticket für Prüfung "Math - Example Exam"

Exam	
Exam Subject	Math
Exam Name	Example Exam

Your Token
33a7d75ef2

Name
Hans Mustermann

Signature

Barcode


PDF

of ticket

This PDF should be printed out and given to the student, when taking the exam (See [Taking an Exam](#)).

1.2.2 Create multiple tickets

A ticket is the access authorization for the student to his/her exam. If you have 20 students to take the exam, you have to generate 20 tickets. You can create a single ticket or multiple at once for a given exam. Create a single ticket by the Actions->Create single ticket wizard and multiple tickets by the Actions->Create multiple tickets wizard.

In this wizard, you can put the names of students in the Names field. You can just copy the names from an external source such as an Excel file or another Office application. The names must be separated by a tab, comma, semicolon, newline or all of them combined. The field tries to read the names as you provide them. How the names are parsed, can be seen in the Preview Proposal on the right of the Names field. You can adjust the names, until the preview is as desired.

Multiple tickets

After pressing Create x tickets, the tickets will be created with default values and you are being redirected to the exam view page. From there you can select Actions->Generate PDFs and you will see a PDF file containing all tickets that are in the open state (See [Ticket states](#)) of the current exam. This can be printed out, and provided to the students when taking the exam (See [Taking an Exam](#)).

1.2.3 The ticket view

All data assigned to a particular ticket are visible in the ticket view. It is divided in different tabs. In this view, it is possible to manage the exam before, during and after. Backups can be browsed and [specific versions of files can be restored](#) during the exam.

General

Here you can see the [state](#), the time at which the exam has started and finished, the duration, whether the ticket is still valid or not (depending on the *Time Limit* you set). The ticket can be assigned to a student by providing his/her name in the *Test Taker* field.

The IP address of the client machine taking the exam is also noted here. To check the connectivity to you can press **Probe**. This will run a check, to see if the client is online or not right now.

Activity Log

This is a log of all client activity on this ticket. You can see every change of the clients state. If a client loses its network connection and reconnects back, it will create a log entry for this event. If a backup fails, you will see a log entry.

Backups

All backup related settings and data can be seen here, including the time of the last successful backup and the last try. Below is a short data breakdown of *every* single backup, that has been performed on this ticket.

Under Actions->Show Log File you can view the whole backup log.

Browse Backup

In this tab you can browse the backup. This is useful, when you have to restore one specific file or directory to a state in the past (See [restore a specific file](#)). Select the date of the backup you want to browse or choose **All version overlapping** if you don't know the date and time of the file you want to find.

All version overlapping means that you will see every file that has existed somewhere along the time period of all backups performed, even if it has been deleted at any time. If you then hover over the file name and click **View all versions**, you will see a list of all versions of that particular file. On the right side, the date and time, its size and permissions are given. **(current)** indicates that this is the current version of the file, as it is in the latest backup. **missing** means that the file has not yet existed or does not anymore exist at that time.

☐ Show hidden files

Path: /home/user/test.odt

Version:

All versions overlapping

test.odt	(current) Jun 21, 2017 12:05:13 PM, 644, 8.10 KiB
test.odt	Jun 19, 2017 9:54:20 AM, 644, 7.84 KiB
test.odt	Jun 19, 2017 9:49:16 AM, <u>missing</u>

Showing 1-3 of 3 items.

browse

Backup

Screenshots

If you configured your exam to take screenshots in an interval, you will see them in this tab. On one hand this is to monitor the student during the exam. On the other hand this can be used to reconstruct the work in progress, if the student forgets to save his/her files. Therefore, it is recommended to activate screenshots and set the interval to a moderate value.

Restores

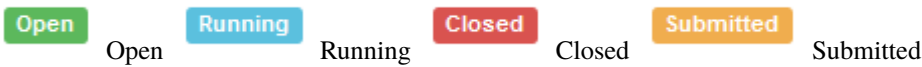
All restore related data can be seen here.

Below is a short data breakdown of *every* single restore, that has been performed on this ticket.

Under **Actions**->**Show Log File** you can view the whole restore log.

1.2.4 Ticket states

Currently there are 4 different states a ticket can occupy. This is indicated in the [ticket view](#) as one of the following batches:



The open state

This indicates that a ticket has not been used yet and there was no activity until now. It is an exam ready to be taken. Newly created tickets are usually in the open state.

The running state

This ticket is in use right now. It indicates an exam being taken by a student at this moment. This means that backups will run on this ticket in the specified interval (all 5 minutes by default). Screenshots (if configured) are taken in the specified interval. The time limit is counting down.

The closed state

A ticket in the closed state means that the exam has finished and the student has pressed **Finish exam** (See [Taking an Exam](#)). It might be possible that the last backup must still be performed in this state, but usually this brands an exam as done.

The submitted state

If the ticket is anonymous (hence no *Test Taker* set) and the exam has finished, the ticket will be in the closed state (see above). On the other hand, if a ticket is assigned to a student by providing a name in the *Test Taker* field and the student finished the exam, the ticket will be in the submitted state. This is just to distinguish between an anonymous and a assigned ticket.

1.3 Taking an Exam

This page describes how to take an exam from the perspective of a student.

The first step is to boot [Lernstick Prüfungsumgebung](#) on your computer from USB. See the following instructions on how to boot your device from USB:

- [Start from USB-Device \(Mac\)](#)
- [Start from USB-Device \(Windows 10\)](#)

As soon as the system has started and you have a working network connection, you can search for the exam server by starting the **Search Exam Server** application.

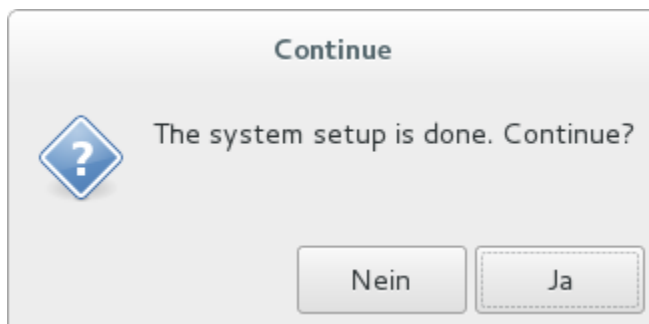


Search Exam Server

If the search was successful, you will be prompted for approval. You will then be prompted for a token. This is the token given on the exam sheet (See [Create a single ticket](#)).

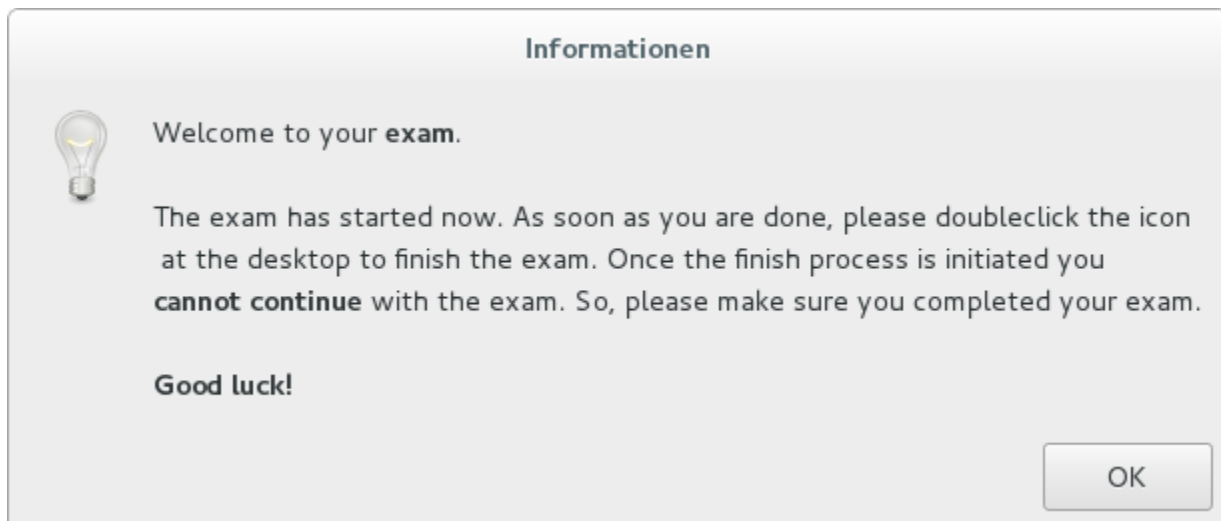
Insert token

If you provided a valid token, the download and preparation of the exam is done and after a while (depending on the size of the exam files) you will see this:



Setup done

Press Yes and the system will immediately restart into the exam (this will take a moment). Finally you will see a desktop and the message below.

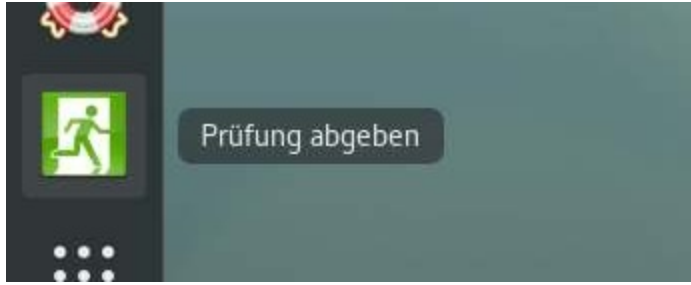


Welcome

to exam

The exam has started now.

To finish the exam click the **Finish** exam icon in the “favorite applications” dash. You should **save all your work** before doing this. **This is not reversible!**



Finish exam 1

This will initiate the last backup of all exam files and marks your exam as finished (See [Ticket states](#)). You will get a notice when the last backup is done, **don't shutdown your computer** before the request message appears. Changes of the exam result from now on, will not be saved.

Finish exam 2

As soon as the message appears, the exam is done. You can shutdown the system now.


1.4 Results

1.4.1 Generate results

Once the exam is done, you might want to get the results for correction in a simple manner. The exam results can be generated into a small zip-file just containing the essential part of the result data. Which part is essential can be configured by you. This can be done in the **Generate results** wizard. Click **Actions->Generate results** to start generating results.

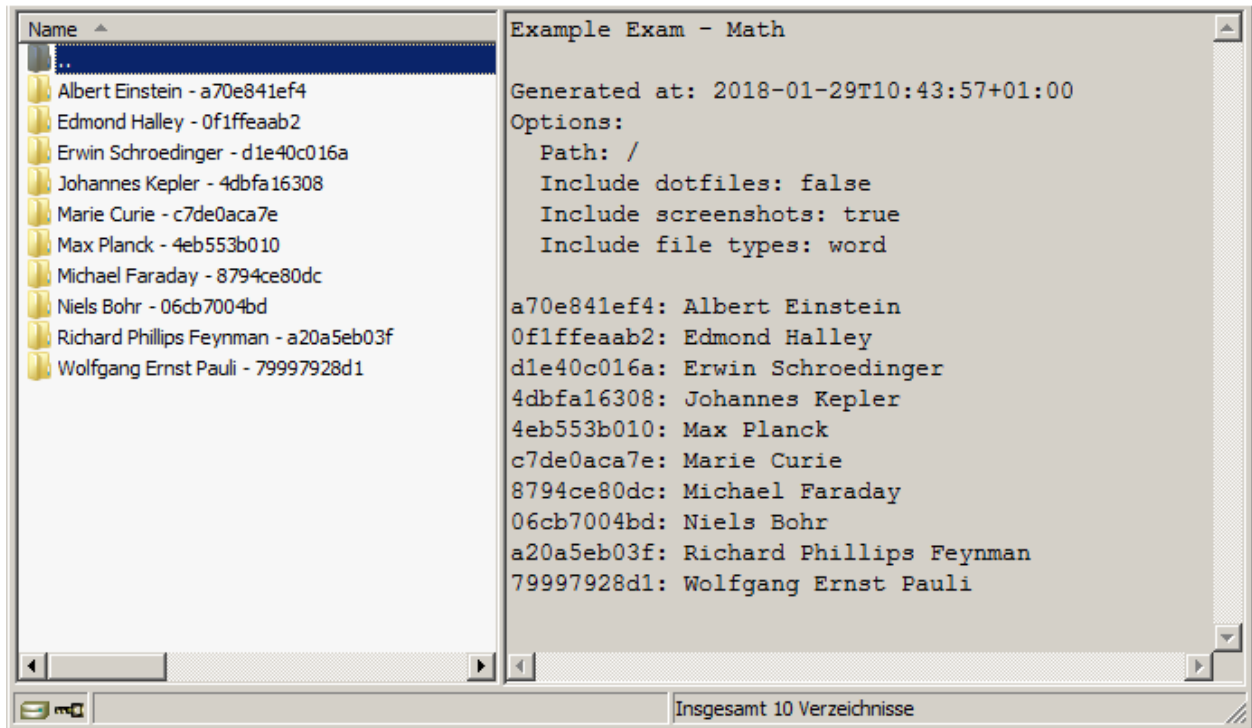
In the first step, you have to choose the *Exam* to generate results for. Proceed to the next step by clicking the **Next step** button below.

In the second step, the configuration is done. Here you can choose which files should be included in the result. Since the backup process just copied ALL files, you can select which of them are relevant to the exam.

For example, if the students only produce text document files, you can select to include only files that match that criteria (by file ending). *Screenshots* can also be included to the result file. *Hidden files* are in most cases not needed to be included. They are normally related to the user profile and will increase the size of result file significantly. For more information about the different settings in the wizard, please click the  Questionmark aside.

If you have advised your students to store their result in a specific directory, you can provide it (relative to the [Remote Backup Path](#)) in the *Path* field.

You can generate the zip-file as often as you want, until it has the desired form. Below is an example on how such a zip file could possibly look like.



Results

file

For every student a separate directory is created. The directory name consists of the students name followed by a dash (-) and then followed by the ticket token. In each directory you will find the files and directories matching the criteria you set in the wizard.

Once you have generated the results, you can save them to your local computer and correct exam results (See [Correct the exams](#)).

1.4.2 Correct the exams

You are completely free in how you correct the exams. If you want to print the files and correct them manually, you can do so. If you want to correct them digitally and hand them back to the student digitally, you can also do so. This howto describes what to do, if you want to go the latter way.

Remember you got a zip-file containing the results (see [Generate results](#)). Unpack file somewhere on your local computer. Correct the exams and save the corrected version in the same directories from the zip-file. After correction, you should have the following directory structure: One directory for each student (Student Name - Token) and inside that directory you should place the corrected exam of the corresponding student. The contents of this directory will be the result that is handed back to the student, so make sure there is no secret content in it.

An example of a digital correction can be seen in the [Example exam: Essay](#).

The next step is to upload the corrected version and [submit the results back to the student](#).

1.4.3 Submit results back to the student

Assuming you [corrected the exams](#) and created a directory structure of the following form:

```
Albert Einstein - a70e841ef4
Edmond Halley - 0f1ffeaab2
Erwin Schroedinger - d1e40c016a
Johannes Kepler - 4dbfa16308
Max Planck - 4eb553b010
Marie Curie - c7de0aca7e
Michael Faraday - 8794ce80dc
Niels Bohr - 06cb7004bd
Richard Phillips Feynman - a20a5eb03f
Wolfgang Ernst Pauli - 79997928d1
```

Note that each directory has the form `Name - Token`, this needs to be satisfied *exactly*. Assuming that in each directory is the corrected exam of the corresponding student and whatever you want to hand back (maybe a sample solution?). Make sure the directories contain only the parts of the corrected exam, you want the student to see. Remove all other data from the directories.

Once you are done, create a new zip-file with the above contents. It then should look like the [generated zip-file](#), just with corrected exams in it.

Start the `Submit results` process now. Click `Actions->Submit results` in the navigation.

Step 1

In Step 1, upload the zip-file you created just now.

After the submitting process, the student will have access to the result as a zip-file likewise. The contents of each zip-file will be everything inside the corresponding directory of the submitted zip-file. So make sure there is no secret content in it.

Click `Next step`.

Step 2

You will see a list like the one below.

#	Test Taker	Token	Exam Name	Notice
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Albert Einstein	a70e841ef4	Example Exam	✓ This Ticket has no submitted result yet.
2	Edmond Halley	0f1ffeaab2	Example Exam	✓ This Ticket has no submitted result yet.
3	Erwin Schroedinger	d1e40c016a	Example Exam	✓ This Ticket has no submitted result yet.
4	Johannes Kepler	4dbfa16308	Example Exam	⚠ There is already a submitted result. The existing one will be overwritten!
5	Marie Curie	c7de0aca7e	Example Exam	✓ This Ticket has no submitted result yet.
6	Max Planck	4eb553b010	Example Exam	⚠ There is already a submitted result. The existing one will be overwritten!
7	Michael Faraday	8794ce80dc	Example Exam	✓ This Ticket has no submitted result yet.
8	Niels Bohr	06cb7004bd	Example Exam	✓ This Ticket has no submitted result yet.
9	Richard Phillips Feynman	a20a5eb03f	Example Exam	✓ This Ticket has no submitted result yet.
10	Wolfgang Ernst Pauli	79997928d1	Example Exam	✓ This Ticket has no submitted result yet.

Submit

step 2

The uploaded zip-file was scanned and the system found the listed entries in it. The green color suggests that everything is ok with those entries in the zip-file. You can also see, that the exam is successfully identified. In the example above, there are 2 tickets with a result submitted already. This is indicated with yellow color. Please notice, that in such a case already existing results will be overwritten permanently. If you want to remove results from being processed, please edit the zip-file and reupload it in Step 1.

Check the list and then press **Submit all results** to submit all the results in the list.

Step 3

The list below shows a short summary on which results could be handed back successfully.

#	Test Taker	Token	Exam Name	Notice
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Albert Einstein	a70e841ef4	Example Exam	✔ Successfully submitted.
2	Edmond Halley	0f1ffeaab2	Example Exam	✔ Successfully submitted.
3	Erwin Schroedinger	d1e40c016a	Example Exam	✔ Successfully submitted.
4	Johannes Kepler	4dbfa16308	Example Exam	✔ Successfully submitted.
5	Marie Curie	c7de0aca7e	Example Exam	✔ Successfully submitted.
6	Max Planck	4eb553b010	Example Exam	✔ Successfully submitted.
7	Michael Faraday	8794ce80dc	Example Exam	✔ Successfully submitted.
8	Niels Bohr	06cb7004bd	Example Exam	⚠ Error submitting result. Please check your zip file.
9	Richard Phillips Feynman	a20a5eb03f	Example Exam	✔ Successfully submitted.
10	Wolfgang Ernst Pauli	79997928d1	Example Exam	✔ Successfully submitted.

Submit

step 3

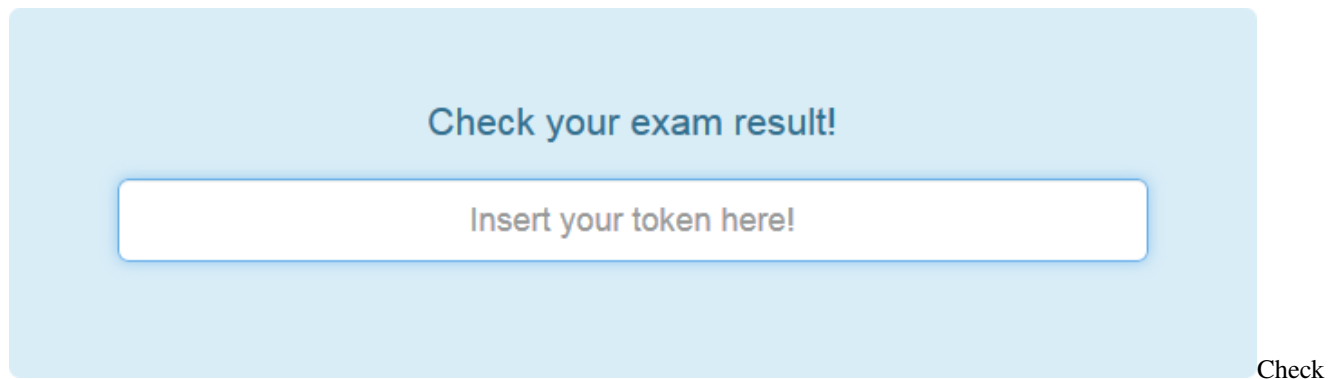
In the example, one exam could not be handed back. This happens mostly when the corresponding directory in the uploaded zip-file is empty, so there is nothing to hand back. In such a case you should check the zip-file again.

That's it. The student can now check his/her exam result.

1.4.4 Get the exam result as a student

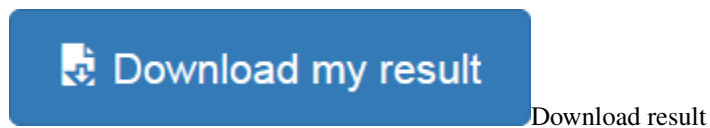
Once the exam result is handed back (See [Submit results](#)) the student can receive the results over the webinterface.

As a student, just browse to the [start](#) page. There is a big blue input field to provide the exam token.



result 1

Once there is a result handed back, the student can download it from here.



If there is no result handed back yet, there will also be no download button.

1.5 Example exam: Essay

This is an example walkthrough of an exam from A to Z. In this case it's a relatively simple exam; an essay. This article covers the whole process:

- creating the exam (including the image file)
- generating tickets (multiple at once with a list of names)
- taking the exam (testing the exam)
- generating results
- submit results back

1.5.1 Goal

The final goal is that the student can write his essay using the [Lernstick-Prüfungsumgebung](#) on his own device using a wireless network. The student should not be allowed to access the internet or any other network resource. The whole exam takes about 3 hours. Once the students handed in their essays, we want to retrieve all essays in a compact form for correction.

This example should also act as a future template for slightly modified or very similar exams.

So, let's get started.

1.5.2 Creating the exam


First we create a new exam, by starting the **Create Exam** wizard under **Actions->Create Exam**.

Next we have to give our exam a *Name* and a *Subject*. I chose **Essay** and **Literature**, because we want to make a literature essay. You are completely free in how you name your exam. This is just for you to identify your exam. You can even use the same name multiple times (though I won't recommend it for the sake of clarity).

The *Time Limit* should be 3 hours, thus **180 minutes**. It must be given in minutes.

The field *Remote Backup Path* will be left as it is, because we don't need to perform a special backup (see [Remote Backup Path](#) for more information on this).

In the **General** section, we only check *Take Screenshots* and set an *Interval* of **1 minute**. All other settings should be left unchecked, since we don't need them.

In the section **Libreoffice**, check *Libreoffice: Save AutoRecovery information* with an *Interval* of **1 minute** and also check *Libreoffice: Always create backup copy*. Those flags cause LibreOffice to save a copy of the document all 1 minutes (to recover the document in case of a crash, even if the student never manually saved the document) and to create a backup copy of the document when saving. For more information, please refer to the  Questionmark aside.

All fields are filled, thus press **Next step**.

Now we need to provide an **Exam File**. Since this is a simple exam (we don't need complex system configuration, such as new applications and special settings), we can use a zip file as exam file, as described in [Create a zip-file as exam file](#).

So we create a zip file on our local computer with the following directory structure:

```
/home/  
/home/user/  
/home/user/Schreibtisch/  
/home/user/Schreibtisch/Hand-in/
```

Then we put in a file named **Essay Topic.pdf** with contents describing the topic of the essay, instructions for the student and where to save the files.

```
/home/user/Schreibtisch/Essay Topic.pdf
```

Make sure, that you also instruct the student to save his/her essay in the directory **Hand-in** placed on the desktop. We created the directory for this purpose. Don't worry, everything the student produces during the exam will be backed up, but later when we [generate the exam results](#), it will be much easier, if all exam results are placed at the same location.

Finally, the zip file is done. Of course you can provide more files and directories if you want.

Now press **Add files...** and upload the created zip file. Once the upload has finished, press **Apply** below.

We have now created an exam, now we need to create tickets for the students to take the exam.

1.5.3 Generating tickets

After pressing **Apply** in the step before, we are now in the **exam view**.


I have a list of students, who should take the exam. Thus, in the exam view, press **Actions->Create assigned Tickets**. This is my example list (though, the essays might be of very high quality):

Ray Bradbury
 James Joyce
 Leo Tolstoy
 Charles Dickens
 J. R. R. Tolkien
 George Orwell
 Jane Austen
 Mary Shelley
 Franz Kafka
 Agatha Christie

However you can copy the names into the **Names** field and see the preview on the right. For more information on creating multiple tickets please refer to [Create multiple tickets](#). Now press **Create 10 Tickets**.

We have now 10 Tickets (with names assigned to them) for our exam. Now print the generated PDFs by clicking **Actions->Generate PDFs**. This will take all tickets in the open state (see [Ticket States](#)) assigned to the exam and produce a PDF file. We now have a 10-paged PDF document with all created tickets in it. Every page has the form as seen in [Create a single ticket](#). Print the document.

When browsing to **Tickets** in the navigation, you should now see the tickets:

#	State	Token	Exam Name	Exam Subject	Started	Finished	Valid	Test Taker
		<input type="text"/>	<input type="text"/>	Literatu 				<input type="text"/>
1	Open	ec76189d07	Essay	Literature	(not set)	(not set)	Yes	Ray Bradbury
2	Open	7a0a1bc610	Essay	Literature	(not set)	(not set)	Yes	James Joyce
3	Open	044240970a	Essay	Literature	(not set)	(not set)	Yes	Leo Tolstoy
4	Open	284dc8a91c	Essay	Literature	(not set)	(not set)	Yes	Charles Dickens
5	Open	2e9e88d110	Essay	Literature	(not set)	(not set)	Yes	J. R. R. Tolkien
6	Open	50b2cf47bb	Essay	Literature	(not set)	(not set)	Yes	George Orwell
7	Open	d14dbefcad	Essay	Literature	(not set)	(not set)	Yes	Jane Austen
8	Open	84015b11e8	Essay	Literature	(not set)	(not set)	Yes	Mary Shelley
9	Open	e0d621cb36	Essay	Literature	(not set)	(not set)	Yes	Franz Kafka
10	Open	0f118bbf49	Essay	Literature	(not set)	(not set)	Yes	Agatha Christie

Showing **1-10** of **10** items.

Ticket

index

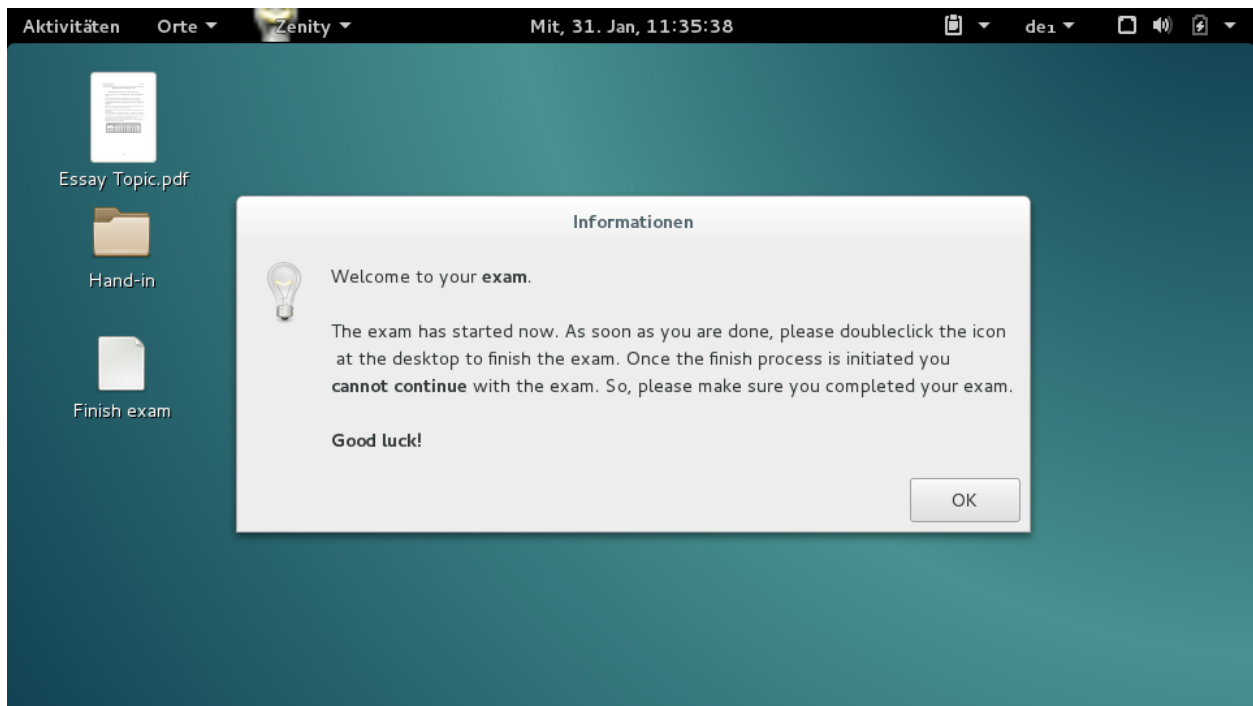
As you can see, all the tickets are in the open state (see [Ticket states](#)) and no start and finish time is set.

We're ready to take the exam.

1.5.4 Taking the exam

Pick one of the tickets and test your exam. This is fully described in [Taking an exam](#), so please refer to this when starting the exam.

From now on I assume you successfully started an exam with one of the tickets. This is how it should look like, after the desktop has loaded.

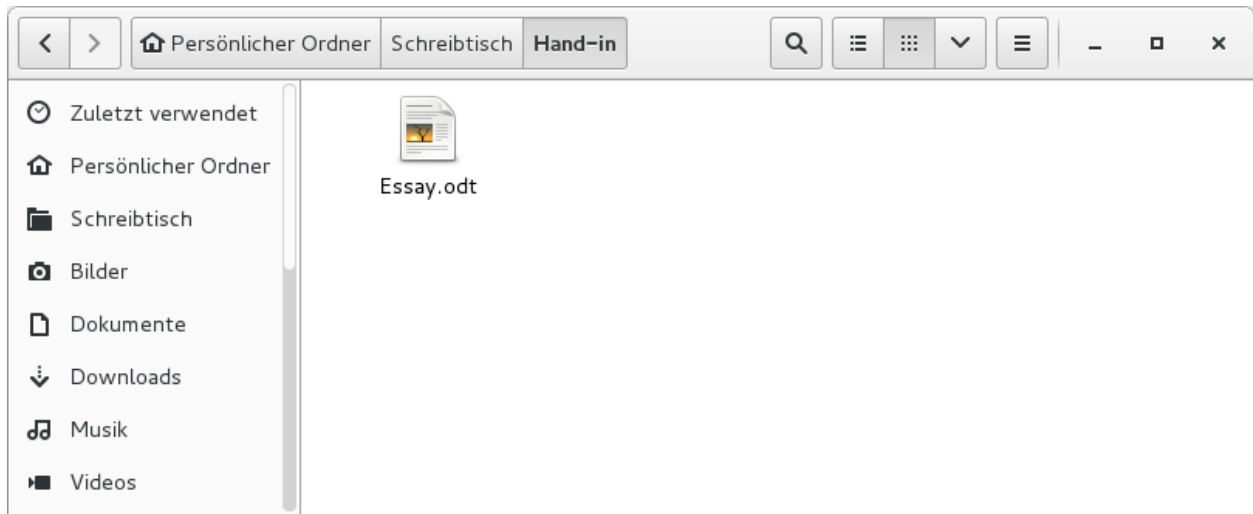
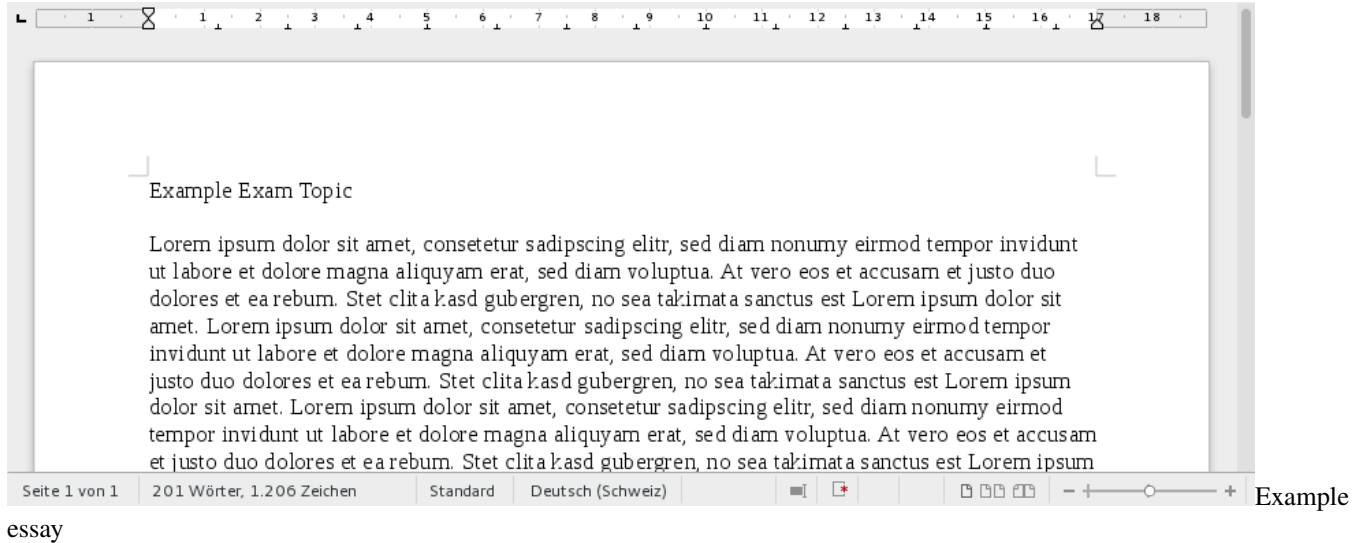


Example

exam

Notice, the `Essay Topic.pdf` file, we placed in the zip file earlier. There is also the `Hand-in` directory and a button `Finish exam` to finish the exam.

We now open LibreOffice Writer, write an example essay and save the file to the `Hand-in` directory.



essay hand-in

Now we finish the exam, by pressing the **Finish exam** button (for details see [Taking an Exam](#)).

We have tested the exam and now we continue by generating the exam results for correction.

1.5.5 Generating exam results


Navigate to **Actions->Generate results**. In the first step we choose the *exam Literature* - *Essay*. We're now asked to configure the resulting zip file with all the exam results in it.

In the *Path* field, type `/Schreibtisch/Hand-in`, because the student was instructed to save all essays documents in the *Hand-in* directory. This will cause the process to only include files and directories from that location.

We also choose to *Include screenshots*.

On the right side, we can even filter further. The exam was an essay, thus the only files produced are **Word documents**. But take care; if you filter too much, you won't be able to extract all relevant data from the results. However we choose **Word documents** here.

In the *Tickets* field, you can select the tickets. The preselected tickets are the ones which are in the closed or submitted state and with no results handed back yet. Only closed or submitted tickets are available to select.

For more information about the fields, please refer to the  Questionmark aside or visit [Generate results](#).

Press now **Generate ZIP-File** to generate the zip-file.

The zip-file will now have the following structure:

```
/Franz Kafka - e0d621cb36/  
/Franz Kafka - e0d621cb36/Screenshots/  
/Franz Kafka - e0d621cb36/Screenshots/screenshot 2018-01-31 11.08.58.jpg  
/Franz Kafka - e0d621cb36/Screenshots/screenshot 2018-01-31 11.09.54.jpg  
(...)  
/Franz Kafka - e0d621cb36/Screenshots/screenshot 2018-01-31 12.05.55.jpg  
/Franz Kafka - e0d621cb36/Essay.odt
```

In our case, we have only one result in the file, thus we have only one directory in the top level. Those directories are always named **Name - Token**. Inside the directory you find the **Screenshots** directory, since we included the screenshots in the step above. For every minute in the exam, we have a screenshot, as configured in the **Create exam** step above. Last but not least, you see the **Essay.odt** file directly in the directory of the student.

If we'd chosen to exclude the screenshots, it would look like this instead:

```
/Franz Kafka - e0d621cb36/  
/Franz Kafka - e0d621cb36/Essay.odt
```

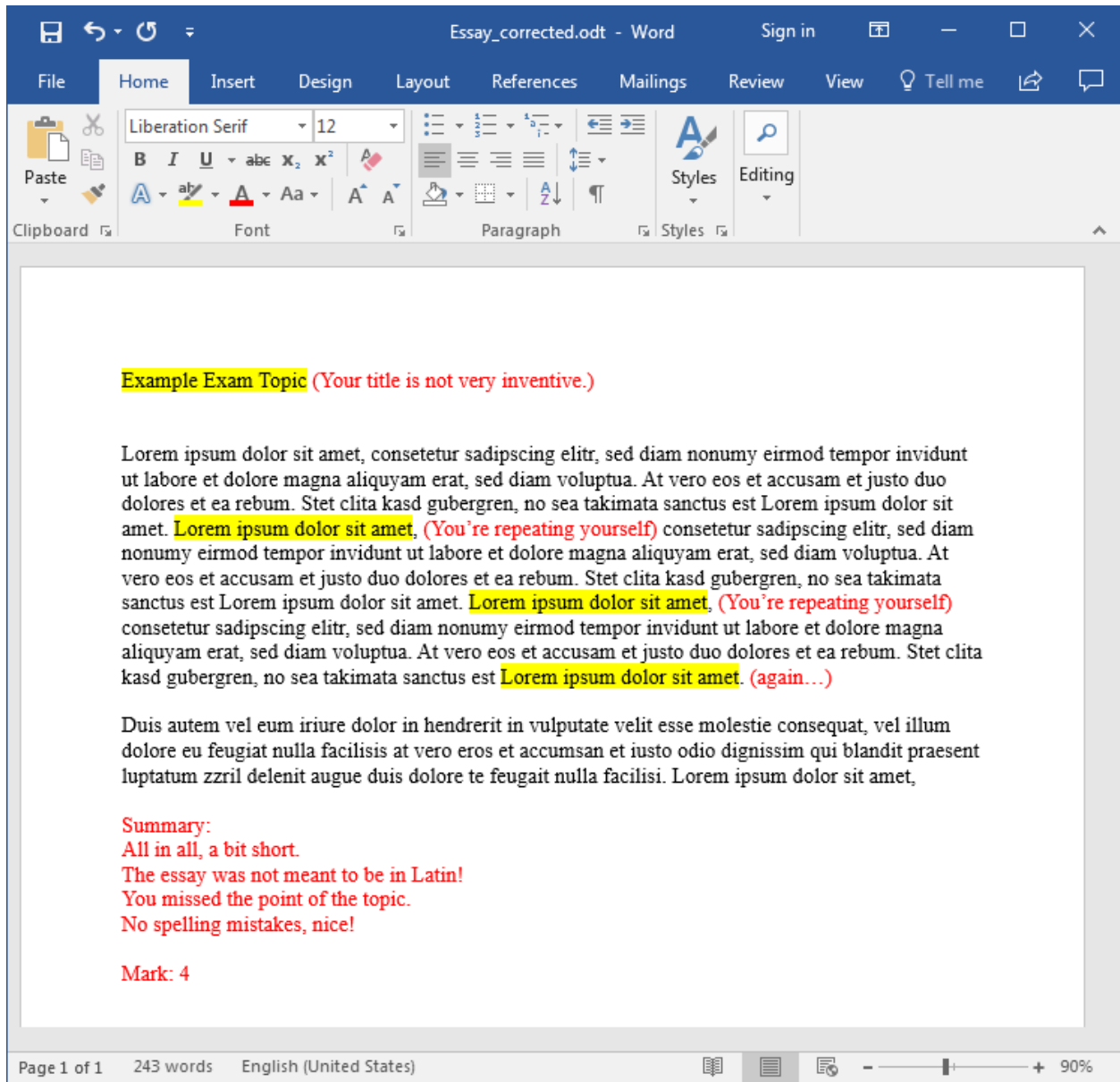
1.5.6 Correcting the results

Now, we have to correct the exam, therefore we extract the zip-file somewhere on our local computer and open the documents.

I chose to copy the original document **Essay_corrected.odt** and write my corrections and the mark in there with red color. You can do that in the way you want to, it doesn't matter.

Of course, you can also print the documents, correct them manually and hand them back physically.

However, I removed the **Screenshots** directory, since I don't want the student to see it in the handed back results. This is an example of a corrected essay:



Corrected

exam

1.5.7 Submit the corrected results back

Again, we create a zip-file with the same directory structure as the results zip-file. So, we have a directory of the form Name - Token for every student. In the corresponding directory are all the corrected files, we want to hand back to the student. My zip-file now looks as follows:

```
/Franz Kafka - e0d621cb36/
/Franz Kafka - e0d621cb36/Essay.odt
/Franz Kafka - e0d621cb36/Essay_corrected.odt
```

Notice, in the example Essay_corrected.odt is the corrected essay (from the picture above), Essay.odt is the original one.

Navigate to Actions->Submit Results to start the Submit results wizard.

Upload the above created zip-file in the first step and press **Next** step. You will see this:

#	Test Taker	Token	Exam Name	Notice
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Franz Kafka	e0d621cb36	Essay	✓ This Ticket has no submitted result yet.

Submit

result

That's all correct. Since we had only one result, there's only one to hand back. So, we continue by pressing **Submit** all results.

#	Test Taker	Token	Exam Name	Notice
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	Franz Kafka	e0d621cb36	Essay	✓ Successfully submitted.

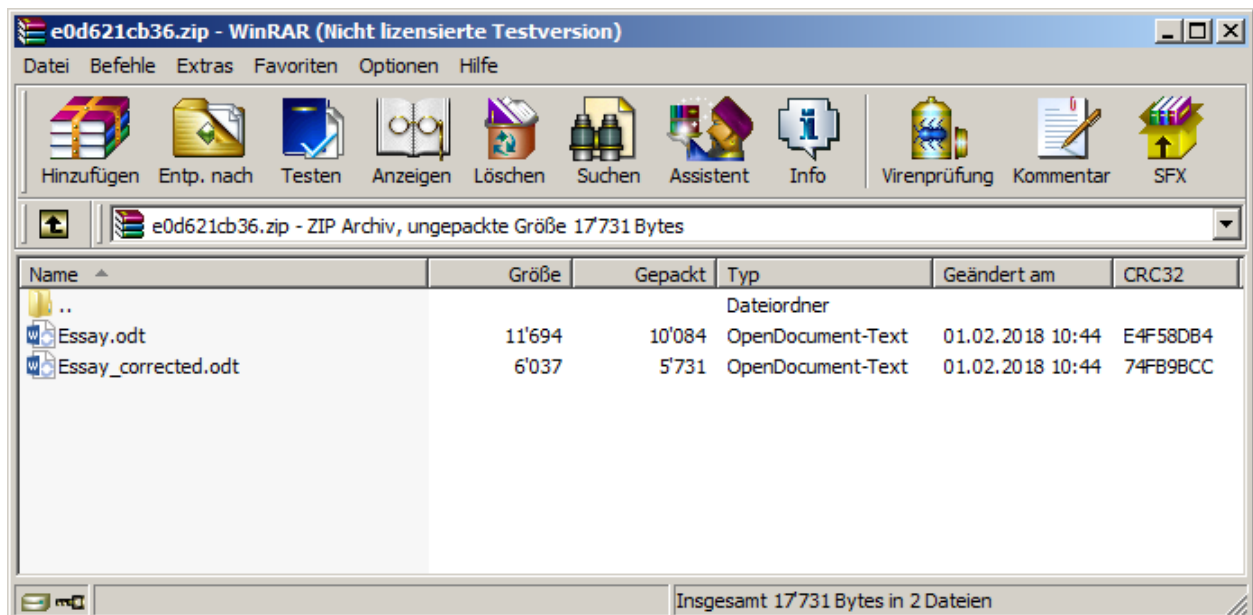
Submit

result done

For detailed information about submitting results back, please refer to [Submit results back to the student](#).

The student can now get the corrected exam results when navigating to the start page and providing the exam token. See [Get the exam result as a student](#) for further information.

The student can now download his/her exam results as a zip-file. In our case, the zip-file will look like this:



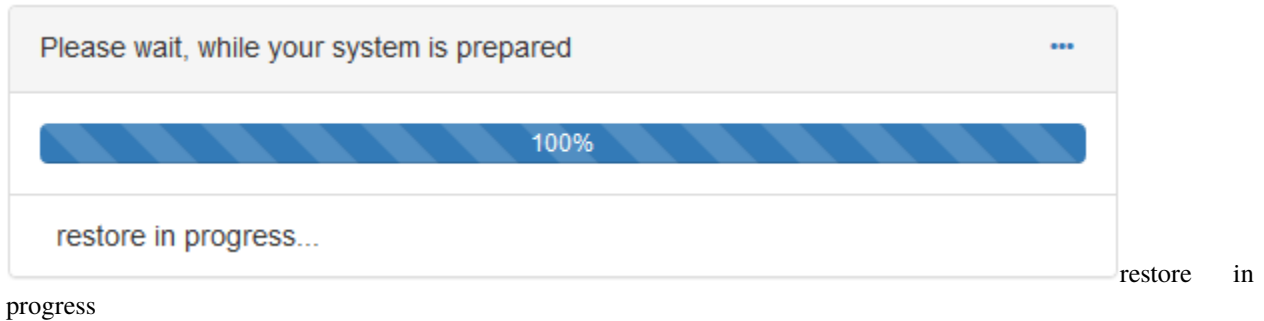
Result

zip

Both, the original document `Essay.odt` and the corrected one `Essay_corrected.odt` with my correction from above are in there.

1.6 Client crashes during exam

If a client has a crash during the exam, you can just restart the machine with an USB-Stick as described in [Taking an Exam](#). Just follow the same steps and you will find the system in the state of the latest backup. **You don't need to manually restore anything**, it will automatically do this itself, while preparing the exam. You may see the restore indicated in the progress bar.



Depending on the *Backup Interval* (see [Create a single ticket](#)), the lost work will be at most once that interval plus the last time the work was saved.

1.7 Restore a specific file

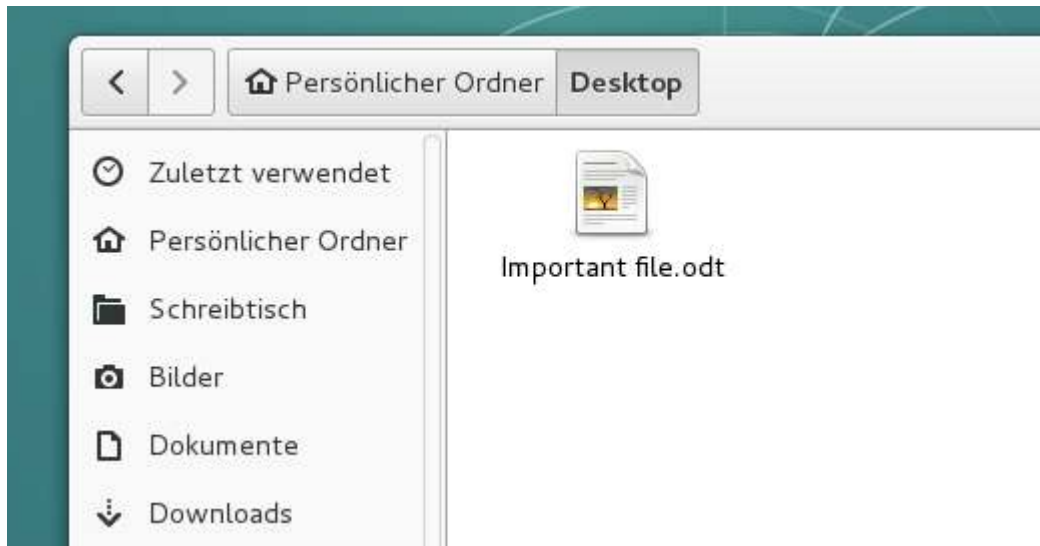
Lets consider one of the students accidentally deleted a file - say `Important file.odt` placed on the Desktop - that is still needed. You can restore any version of the file, which is covered by one of the past backups.

1. Go to the [Ticket view](#) of the affected ticket.
2. Switch to the `Browse Backup` tab.
3. Select `All versions overlapping` in the version selection on the right side, so you can see every file that has existed in any point of the history.
4. Browse to the location where the file is missing (`Desktop` in this example). You can check the full path on the top left side.
5. If the file has been present in one of the backup instances, you will now see the file.
6. Hover over the filename and click `View all versions` to get a view of all versions available in the backup.

Restore file 1

7. On the right side of each filename, you see the backup age, the file permissions and its size. The word `current` indicates the latest (current) version of the file. You can click on the filename to directly download the file to your local computer and check if this is the version you want to restore. A crossed out filename means that the file has not existed in that backup or the ones before. If you restore a file in the crossed state, it will be deleted on the target machine.
 8. Once you found the desired version of the file, you can click `Restore this state of the file`.
 9. Confirm the restore after carefully reading the warnings.
- Restore file 2
10. Wait for the restore to complete.
 11. You can check the restore progress and log in the `Restores` tab.
 12. On the students computer, browse to the location, where the file was missing and press F5 to reload the directory contents.

13. The file should now be present in the restored version.



Restore file 3

1.8 Hardware Recommendations

The hardware recommendations are given in terms of CPUs, memory and disk space.

1.8.1 Number of CPUs

Please consult the following table, where n = (number of concurrent exams you wish to perform):

1.8.2 RAM

The required memory by GLaDOS can be calculated by the following formula:

$\begin{aligned} \text{Total Memory} &= (\text{maximum number of running daemons}) * (32\text{MB}) + 120\text{MB} \\ &+ (\text{number of concurrent exams you wish to perform}) * (5\text{MB}) + 240\text{MB} \\ &+ 1000\text{MB} \end{aligned}$

- Each daemon needs roughly 32MB of unshared memory. The amount of shared memory is approximately 120MB.
- Each event stream or agent needs roughly 5MB of unshared memory. The amount of shared memory is approximately 240MB.

1.8.3 Storage

The amount of disk storage needed by GLaDOS highly depends on various parameters:

- How long you wish to keep backups.
- How many data do students produce on average during an exam.
- How long an average exam will take.
- The setting of [Remote Backup Path](#).
- The setting of the screenshots interval.
- Whether you activate [Screen Capturing](#) or not.

Make sure that there is enough disk storage for your needs.

1.9 Installation Guide (From the sources)

This guide describes how to install GLaDOS from the source package.

1.9.1 Requirements

Webserver

Since GLaDOS's interface is based entirely in your browser, you'll need a web server (such as Apache, nginx) to install the files into.

- Apache needs [mod_rewrite](#) to be installed and enabled.
- Nginx needs PHP as an [FPM SAPI](#).

PHP

- You should have PHP 5.4 or above. Ideally latest PHP 7. You should also install the [PDO PHP Extension](#) and the corresponding database driver `pdo_mysql`.
- To support generating of ZIP files, you need the PHP `zip` extension.
- To support live data, you need the [PECL inotify](#) extension version 0.1.6.
- To support creating thumbnails, you need the PHP `gd` extension.
- Yii2 needs the Multibyte-Strings [mbstring](#) module for PHP.

Database

Currently only MySQL databases are supported. You need MySQL 5.5 or newer.

Miscellaneous

- To fetch the exam backups, [rdiff-backup](#) 1.2.8 or newer is needed.
- [OpenSSH](#) client and its key generator `ssh-keygen` are needed to create a connection for rdiff-backup.
- To support squashfs files, [Squashfs](#) is needed.
- To support auto discovery of the exam server, [avahi](#) is needed.
- For the manual installation [Composer](#) is needed. It can be removed subsequently.

Debian

In Debian, the packages needed from above can be installed by:

```
apt-get install apache2 mysql-server php5 php5-mysql php5-gd squashfs-tools rdiff-backup_
↪avahi-daemon openssh-client
```

1.9.2 Installation

Composer can be installed to `/usr/local/bin` with the following commands (you need [curl](#) for this to work):

```
cd /usr/src
curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --
↪filename=composer
```

Download the latest source package

Browse to the Github [release page](#) and download the latest version of GLaDOS.

```
curl -L -O https://github.com/imedia/glados/archive/$version.tar.gz
```

Where `$version` is the latest version number.

Unpack the source package:

```
tar xfs $version.tar.gz
```

Create a new directory `/usr/share/glados` and copy all extracted files. Then `cd` into the created directory:

```
mkdir /usr/share/glados
cp -rpv glados-1.0.3/* /usr/share/glados/
cd /usr/share/glados
```

Install composer asset plugin (according to the [Yii2 installation guide](#)):

```
composer global require "fxp/composer-asset-plugin:^1.4.1"
```

Run composer (this will take a while, you also may need to create a [Github OAuth token](#) to go over the [API rate limit](#)):

```
composer update
```

This will install all packages specified in `composer.json`.

Create the following directories:

```
mkdir -p /var/lib/glados/uploads
mkdir -p /var/lib/glados/backups
mkdir -p /var/lib/glados/results
mkdir -p /var/lib/glados/tmp
mkdir -p /var/lib/glados/.ssh
mkdir -p /var/log/glados
```

Some directories need to be writable by the user under which your webserver runs. Assuming the user is called `www-data`, adjust the following permissions:

```
chown www-data /var/log/glados
chown www-data:www-data /usr/share/glados/web/assets/
chown -R www-data:www-data /var/lib/glados
```

Next, set the environment variables. Edit `/usr/share/glados/web/index.php` and comment out lines 4 and 5:

```
// comment out the following two lines when deployed to production
//defined('YII_DEBUG') or define('YII_DEBUG', true);
//defined('YII_ENV') or define('YII_ENV', 'dev');
```

MySQL setup

Once all dependencies are installed, you need to set up the database. The following example code, shows how to create a database called `glados`, with a database user called `glados` and password `mysqlpassword` with all permissions granted on that database.

```
mysql -u root -p
Enter password:
mysql> create database glados;
mysql> grant usage on *.* to glados@localhost identified by 'mysqlpassword';
mysql> grant all privileges on glados.* to glados@localhost;
mysql> quit
```

Open the config file `/usr/share/glados/config/db.php` and provide the database name, the username and the password from above (see [Config Files](#) for more information):

```
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=glados',
    'username' => 'glados',
    'password' => 'mysqlpassword',
    'charset' => 'utf8',
];
```

Change into the directory `/usr/share/glados`. Install all RBAC tables:

```
./yii migrate --migrationPath=@yii/rbac/migrations --interactive=0
```

RBAC initialization:

```
./yii rbac/init --interactive=0
```

Database tables:

```
./yii migrate --interactive=0
```

PHP setup

Install PECL inotify (you need [PEAR](#), PHP5 module development files and a C-compiler for this to work):

```
pecl install inotify          //for PHP7
pecl install inotify-0.1.6    //for PHP5
```

Create a PHP ini file for inotify (example: `/etc/php/7.0/mods-available/inotify.ini` for PHP7 and `/etc/php5/mods-available/inotify.ini` for PHP5) with contents:

```
; configuration for php inotify module
; priority=20
extension=inotify.so
```

Make sure to enable the inotify module.

Avahi setup

Create an Avahi service file (`/etc/avahi/services/glados.service`) with contents:

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">Glados on %h</name>
  <service>
    <type>_http._tcp</type>
    <port>80</port>
    <txt-record>type=Glados</txt-record>
    <txt-record>actionDownload='glados/index.php/ticket/download/{token}'</txt-record>
    <txt-record>actionFinish='glados/index.php/ticket/finish/{token}'</txt-record>
    <txt-record>actionNotify='glados/index.php/ticket/notify/{token}?state={state}'</txt-
    record>
    <txt-record>actionSSHKey='glados/index.php/ticket/ssh-key'</txt-record>
    <txt-record>actionMd5='glados/index.php/ticket/md5/{token}'</txt-record>
    <txt-record>actionConfig='glados/index.php/ticket/config/{token}'</txt-record>
  </service>
</service-group>
```

Make sure that the file above contains the correct URLs (this depends on your webserver setup, see below). For example the download URL `actionDownload` will be made up of the hosts IP-address, the port, the protocol and the given relative path from the txt-record `actionDownload`, discovered by `avahi-browse`. The protocol is determined by the port number, `80` gives `http` and `443` gives `https`.

```
${gladosProto}://${gladosIp}:${gladosPort}/${actionDownload}
```

will then be

```
http://1.2.3.4:80/glados/index.php/ticket/download/{token}
```

If you use the Apache setup from below, you don't have to change the service file.

Finally restart the avahi-daemon:

```
/etc/init.d/avahi-daemon restart
```

Webserver setup

Apache

Use the following configuration in Apache's `httpd.conf` file or within a virtual host configuration (example: `/etc/apache2/conf-available/glados.conf`).

```
Alias /glados /usr/share/glados/web

<Directory /usr/share/glados/web>
    Options FollowSymLinks
    DirectoryIndex index.php

    # use mod_rewrite for pretty URL support
    RewriteEngine on
    # If a directory or a file exists, use the request directly
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    # Otherwise forward the request to index.php
    RewriteRule . index.php

    # ...other settings...
</Directory>
```

Make sure `mod_rewrite` is enabled and installed.

Then restart apache:

```
/etc/init.d/apache2 restart
```

Nginx

To use Nginx, you should install PHP as an **FPM SAPI**. You may use the following Nginx configuration, replacing `glados.test` with the actual hostname to serve.

```
server {
    charset utf-8;
    client_max_body_size 128M;

    listen 80; ## listen for ipv4
```

(continues on next page)

(continued from previous page)

```

#listen [::]:80 default_server ipv6only=on; ## listen for ipv6

server_name glados.test;
root        /usr/share/glados/web;
index       index.php;

access_log  /var/log/glados/access.log;
error_log   /var/log/glados/error.log;

location / {
    # Redirect everything that isn't a real file to index.php
    try_files $uri $uri/ /index.php$is_args$args;
}

# uncomment to avoid processing of calls to non-existing static files by Yii
#location ~ \.(js|css|png|jpg|gif|swf|ico|pdf|mov|fla|zip|rar)$ {
#    try_files $uri =404;
#}
#error_page 404 /404.html;

# deny accessing php files for the /assets directory
location ~ ^/assets/.*\.php$ {
    deny all;
}

location ~ \.php$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    #fastcgi_pass 127.0.0.1:9000;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    try_files $uri =404;
}

location ~* /\. {
    deny all;
}
}

```

When using this configuration, you should also set `cgi.fix_pathinfo=0` in the `php.ini` file in order to avoid many unnecessary system `stat()` calls.

Also note that when running an HTTPS server, you need to add `fastcgi_param HTTPS on;` so that Yii can properly detect if a connection is secure.

Check the installation

Hit <http://localhost/glados/requirements.php> to check if all requirements are met.

Make sure you have set `upload_max_filesize` and `post_max_size` to a proper value in `php.ini`.

After checking all requirements, you can remove the `requirements.php` file located at `/usr/share/glados/web/requirements.php`.

You can now access the webinterface by the URL <http://localhost/glados>.

You may login with **admin/admin** or **teacher/teacher**. To modify the users, please login as **admin**.

```
:wq
```

1.10 Installation Guide (Debian package)

This guide describes how to install GLaDOS from the Debian package.

1.10.1 Debian 9/10

Download the newest packages from the Github [releases page](#).

Download both `glados_<glados_version>-deb9_all.deb` and `yii2-glados_<yii_version>_all.deb` to your server. Simply install the two packages using the following command inside the directory of the downloaded files (substituting the corresponding `<yii_version>` and `<glados_version>` you've just downloaded):

```
apt install ./yii2-glados_<yii_version>_all.deb ./glados_<glados_version>-deb9_all.deb
```

During the installation you will be asked some questions (You should note down these settings):

- *Configure database for glados with dbconfig-common?:* **yes**
- *MySQL password for glados:* **<your_desired_password>**
- *Repeat password:* **<your_desired_password>**

If this fails due to dependency issues, install the needed dependencies and complete the installation by running:

```
apt-get -f install
```

You can now access the webinterface by the URL <http://localhost/glados>.

You may login with **admin/admin** or **teacher/teacher**. To modify the users, please login as **admin**.

1.10.2 Debian 8

Requirements

To install the needed requirements, run the following command:

```
apt-get install apache2 mysql-server php5 php5-mysql php5-gd squashfs-tools rdiff-backup_
↪avahi-daemon openssh-client dbconfig-common
```

Installation

Get the newest packages from the Github [releases page](#).

Install the packages in the following order (for example version 1.0.3):

```
dpkg -i php5-inotify_0.1.6-1_amd64.deb
dpkg -i yii2-glados_2.0.13.1-1_all.deb
dpkg -i glados_1.0.3-1_all.deb
```

You can now access the webinterface by the URL <http://localhost/glados>.

You may login with **admin/admin** or **teacher/teacher**. To modify the users, please login as **admin**.

1.11 Softwareupdate on Debian

This guide describes how to update GLaDOS on a Debian server using the deb-packages.

If you plan to update GLaDOS version $\leq 1.0.4$ on a Debian 8 server, please notice that GLaDOS version 1.0.5 and onwards will **not support Debian 8 anymore**. Please consider an upgrade of Debian 8 to Debian 9 following this [Guide](#).

1.11.1 Updating GLaDOS

To update GLaDOS, simply download the desired release version on [GitHub](#). Download both `glados_<glados_version>-deb9_all.deb` and `yii2-glados_<yii_version>_all.deb` to your server.

First update the local package information list:

```
apt update
```

Install all available updates:

```
apt upgrade
```

Install the newest version by executing (inside the directory of the downloaded files):

```
apt install ./yii2-glados_<yii_version>_all.deb ./glados_<glados_version>-deb9_all.deb
```

During the update you may be asked some questions:

- Configuration file `'/etc/glados/params.php'` ==> Modified (by you or by a script) since installation.: Y

```

Configuration file '/etc/glados/params.php'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
     Y or I  : install the package maintainer's version
     N or O  : keep your currently-installed version
     D       : show the differences between the versions
     Z       : start a shell to examine the situation
   The default action is to keep your current version.
*** params.php (Y/I/N/O/D/Z) [default=N] ? Y

```

- *Upgrade database for glados with dbconfig-common?: yes*

You might have to adjust settings in config files you have done so far (see [Glados config files](#)). Your existing installation will be updated and all needed dependencies are automatically installed.

1.12 Upgrade from Debian 8 to 9

This guide describes how to upgrade GLaDOS from Debian 8.x to Debian 9.x with all data migrated. The old Debian 8 server will be called `old8` and the new Debian 9 server will be called `new9` throughout the whole guide.

1.12.1 Preliminaries

These steps have to be done on `old8`.

First, make a backup of all data on the Debian 8 server.

Stop all running daemons via the webinterface.

Upgrade GLaDOS on your server running Debian 8 to version 1.0.4 (see [Softwareupdate on Debian](#)).

Create a dump of the MySQL-database of GLaDOS:

```
mysqldump -u 'glados_user' -p 'glados_database' > /tmp/glados_db_1.0.4.sql
```

Notice that, `glados_user` and `glados_database` must be adjusted according to your setup. You can find the username and database name in the file `/etc/glados/config-db.php`. The contents of that file could look for example:

```

<?php
##
## database access settings in php format
## automatically generated from /etc/dbconfig-common/glados.conf
## by /usr/sbin/dbconfig-generate-include
## Tue, 31 Jul 2018 08:47:43 +0200
##
## by default this file is managed via ucf, so you shouldn't have to
## worry about manual changes being silently discarded. *however*,
## you'll probably also want to edit the configuration file mentioned
## above too.
##
$dbuser='user';

```

(continues on next page)

(continued from previous page)

```
$dbpass='secret';  
$basepath='';  
$dbname='glados';  
$dbserver='';  
$dbport='';  
$dbtype='mysql';
```

In the above case you would have to run the following command:

```
mysqldump -u 'user' -p 'glados' > /tmp/glados_db_1.0.4.sql
```

There will be a prompt asking for the password. In this case the requested password is **secret** (without the single quotes).

Copy the file `/tmp/glados_db_1.0.4.sql` somewhere you have access to for later.

1.12.2 Setup of the new Debian 9 server

These step have to be done on new9.

On the new Debian 9 server, install a fresh version of GLaDOS 1.0.4 (see [GitHub releases](#)) according to the [Installation Guide](#). It is important that you install exactly GLaDOS version **1.0.4** and not another one. Note down the **password** you have provided during the installation for later use.

1.12.3 Data migration (files)

These step have to be done on new9.

Once it's installed, you can start copying the files from the old server to the new one. Directories you have to copy:

- `/var/lib/glados/uploads`
- `/var/lib/glados/backups`
- `/var/lib/glados/results`
- `/var/lib/glados/.ssh`
- `/var/log/glados`

If you have changed these directories you have to copy the directories configured in `/etc/glados/params.php` (see [Glados config files](#) for more details), namely:

- `uploadPath`
- `backupPath`
- `resultPath`
- `dotSSH`
- `daemonLogFilePath`

You can use `rsync` to transfer the files (the `/` at the end of the directory name is important!):

```
rsync -arvpgot -e ssh --delete root@old8:/var/lib/glados/uploads/ /var/lib/glados/uploads  
rsync -arvpgot -e ssh --delete root@old8:/var/lib/glados/backups/ /var/lib/glados/backups  
rsync -arvpgot -e ssh --delete root@old8:/var/lib/glados/results/ /var/lib/glados/results
```

(continues on next page)

(continued from previous page)

```
rsync -arvpgot -e ssh --delete root@old8:/var/lib/glados/.ssh/ /var/lib/glados/.ssh
rsync -arvpgot -e ssh --delete root@old8:/var/log/glados/ /var/log/glados
```

You can also copy the data with another method, but make sure that the **permission, ownership, group and the modification time** of the files are preserved (the above `rsync` commands do take this into account).

1.12.4 Data migration (database)

These step have to be done on `new9`.

Copy the database dump file from the old Debian 8 server to the `/tmp` directory of the new Debian 9 server:

```
scp -p root@old8:/tmp/glados_db_1.0.4.sql /tmp
```

Connect to the database server:

```
mysql -u glados -p
```

Provide the **password** you have noted down during the installation here.

In the terminal, you should now see a prompt like

```
MariaDB [(none)]>
```

Switch to the `glados` database by issuing

```
use glados;
```

The dumped database from earlier in this guide can now be restored by

```
source /tmp/glados_db_1.0.4.sql;
```

Close the console by

```
quit
```

1.12.5 Upgrade of the new server

You are now ready to update GLaDOS on the new Debian 9 server to the newest version according to the [Guide on Softwareupdate on Debian](#).

1.13 Upgrade from Debian 9 to 10

This guide describes how to upgrade GLaDOS from Debian 9.x to Debian 10.x with all data migrated. The old Debian 9 server will be called `old9` and the new Debian 10 server will be called `new10` throughout the whole guide.

1.13.1 Preliminaries

These steps have to be done on old9.

First, perform a backup of all data on the Debian 9 server.

Stop all running daemons via the webinterface.

Upgrade GLaDOS on your server running Debian 9 to version 1.0.11 (see [Softwareupdate on Debian](#)).

Create a dump of the MariaDB-database of GLaDOS:

```
mysqldump -u 'glados_user' -p 'glados_database' > /tmp/glados_db_1.0.11.sql
```

Notice that, `glados_user` and `glados_database` must be adjusted according to your setup. You can find the username and database name in the file `/etc/glados/config-db.php`. The contents of that file could look for example:

```
<?php
##
## database access settings in php format
## automatically generated from /etc/dbconfig-common/glados.conf
## by /usr/sbin/dbconfig-generate-include
## Tue, 31 Jul 2018 08:47:43 +0200
##
## by default this file is managed via ucf, so you shouldn't have to
## worry about manual changes being silently discarded. *however*,
## you'll probably also want to edit the configuration file mentioned
## above too.
##
$dbuser='user';
$dbpass='secret';
$basepath='';
$dbname='glados';
$dbserver='';
$dbport='';
$dbtype='mysql';
```

In the above case you would have to run the following command:

```
mysqldump -u 'user' -p 'glados' > /tmp/glados_db_1.0.11.sql
```

There will be a prompt asking for the password. In this case the requested password is `secret` (without the single quotes).

Copy the file `/tmp/glados_db_1.0.11.sql` somewhere you have access to for later.

1.13.2 Setup of the new Debian 10 server

These steps have to be done on new10.

On the new Debian 10 server, install a fresh version of GLaDOS 1.0.11 (see [GitHub releases](#)) according to the [Installation Guide](#). It is important that you install exactly GLaDOS version **1.0.11** and not another one. Note down the **password** you have provided during the installation for later use.

1.13.3 Data migration (files)

These step have to be done on new10.

Once it's installed, you can start copying the files from the old server to the new one. Directories you have to copy:

- /var/lib/glados/uploads
- /var/lib/glados/backups
- /var/lib/glados/results
- /var/lib/glados/sc
- /var/lib/glados/.ssh
- /var/log/glados

If you have changed these directories you have to copy the directories configured in `/etc/glados/params.php` (see [Glados config files](#) for more details), namely:

- uploadPath
- backupPath
- resultPath
- scPath
- dotSSH
- daemonLogFilePath

You can use `rsync` to transfer the files (the `/` at the end of the directory name is important!):

```
rsync -arvpgot -e ssh --delete root@old9:/var/lib/glados/uploads/ /var/lib/glados/uploads
rsync -arvpgot -e ssh --delete root@old9:/var/lib/glados/backups/ /var/lib/glados/backups
rsync -arvpgot -e ssh --delete root@old9:/var/lib/glados/results/ /var/lib/glados/results
rsync -arvpgot -e ssh --delete root@old9:/var/lib/glados/sc/ /var/lib/glados/sc
rsync -arvpgot -e ssh --delete root@old9:/var/lib/glados/.ssh/ /var/lib/glados/.ssh
rsync -arvpgot -e ssh --delete root@old9:/var/log/glados/ /var/log/glados
```

You can also copy the data with another method, but make sure that the **permission, ownership, group and the modification time** of the files are preserved (the above `rsync` commands do take this into account). The `--delete` flag deletes extraneous files from destination directories on new10.

1.13.4 Data migration (config)

These step have to be done on new10.

If you included authentication methods, such as Active Directory authentication, copy the configuration of the authentication methods to the new system:

```
scp -p root@old9:/etc/glados/auth*.php /etc/glados/
```

1.13.5 Data migration (database)

These step have to be done on new10.

Copy the database dump file from the old Debian 9 server to the /tmp directory of the new Debian 10 server:

```
scp -p root@old9:/tmp/glados_db_1.0.11.sql /tmp
```

Connect to the database server:

```
mysql -u glados -p
```

Provide the **password** you have noted down during the installation here.

In the terminal, you should now see a prompt like

```
MariaDB [(none)]>
```

Switch to the glados database by issuing

```
use glados;
```

The dumped database from earlier in this guide can now be restored by

```
source /tmp/glados_db_1.0.11.sql;
```

Close the console with

```
quit
```

1.13.6 Data migration (LDAPS)

These step have to be done on new10.

If you use LDAP authentication with SSL (see [LDAP with SSL](#)), you may have to import the LDAP servers CA certificate into the certificate store of the new server. To import all additional certificates from the old server to the new one, run the following command on the new server:

```
scp -p root@old9:/usr/local/share/ca-certificates/your_certificate.crt /usr/local/share/  
↪ca-certificates/
```

where you have to replace `your_certificate.crt` with your CA certificate. After that, update the CA store:

```
update-ca-certificates -v
```

In order to make sure that everything works as intended, you should [test](#) the authentication process on the new server.

1.13.7 Upgrade of the new server

You are now ready to update GLaDOS on the new Debian 10 server from version 1.0.11 to the newest version according to the Guide on [Softwareupdate on Debian](#).

1.14 Glados config files

The main configuration can be found in the file `config/params.php`. This is a list of the config settings and what they mean:

Settings with a star * are now accessible for admin users over `System->Settings` and should not be changed in the `params.php` file anymore, since the value from `System->Settings` takes precedence over the file.

The database connection can be configured in the file `config/db.php`, which usually looks like this:

```
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=glados',
    'username' => 'glados',
    'password' => 'mysqlpassword',
    'charset' => 'utf8',
];
```

This list describes the meanings of the keys:

1.15 Exam client configuration

To setup an exam client, you can install the package `lernstick-exam-client` on a Lernstick:

```
apt-get install lernstick-exam-client
```

You can also download the Debian Package from the [Github releases page](#).

The exam client can be configured in 2 ways.

1.15.1 Autodiscovery of the exam server

To configure the exam client for this behavior, nothing has to be done. After installing the package, you can just search for the exam server, as described in [Taking an Exam](#).

Notice, for this to work, you have to [configure the network](#) accordingly.

1.15.2 Exam server with fixed IP-address

If you want a more secure installation and the exam server has a fixed IP-address, you can configure the exam client to only access your fixed exam server.

Create a config file `/etc/lernstick-exam-client.conf` with the following contents:

```
gladosIp="1.2.3.4"
gladosHost="examsrv"
gladosPort=80
gladosProto="http"
gladosDesc="Description"

actionDownload='glados/index.php/ticket/download/{token}'
actionFinish='glados/index.php/ticket/finish/{token}'
actionNotify='glados/index.php/ticket/notify/{token}?state={state}'
actionSSHKey='glados/index.php/ticket/ssh-key'
actionMd5='glados/index.php/ticket/md5/{token}'
actionConfig='glados/index.php/ticket/config/{token}'
```

The below table explains the keys and values:

If you start the `Search Exam Server` utility now, it will only search for your given IP-address, thus other exam servers in the network will be ignored.

1.15.3 Automatically search for exam server

You can configure the exam client, so that on **every** network connection that comes up, the search for an exam server is started. This can be done by setting

```
SearchExamServer=true
```

in the config file `/etc/lernstickWelcome`.

Notice, that the utility now starts on **every** network connection (LAN, WLAN, VPN and so on).

It is recommended to create a shortcut for `Search Exam Server` on the Desktop.

1.16 Network configuration

There are 2 possible ways you can configure your setup. You can either configure the network in the way that the client can autodiscover the exam server, or configure the clients to access a fixed IP-address. The client setup can be found [here](#).

1.16.1 Autodiscovery of the exam server

For this to work, you need to be able to discover the exam server via [Bonjour](#). Therefore DNS Service Discovery (DNS-SD) must be allowed in the network. It is necessary to strictly allow this in your network. If you use a wireless network as exam network, notice that most accesspoints will disable this by default. If your server sits in another subnet, you have to route the needed ports, according to the Bonjour standard. Bonjour clients will talk via UDP port 5353 and multicast IP-address 224.0.0.251 (IPv4), ff02::fb (IPv6) respectively.

Avahi includes several utilities which help you discover the services running on a network. For example, run

```
avahi-browse -r --no-db-lookup _http._tcp
```

to discover services in your network. If your network and server are configured appropriately, you should discover the exam server with the above command.

1.16.2 Exam server with fixed IP-address

This is a more secure setup. To run the exam server in a network, you have to make sure that clients can connect to the server. On the other hand, the server needs to be able to connect to the clients.

This is a list of services and ports that must be allowed:

If your exam server sits in another subnet than your exam clients, make sure to disable [Network address translation](#) in the configuration of the router. This is necessary because the server detects the clients IP-address (which is then used in remote backup) over the HTTP headers and/or `$_SERVER` environment variables provided by the webserver. With NAT enabled the header would contain the IP-address of the routing device instead.

1.17 GLaDOS System Settings

There are currently a few global configuration settings that can be modified by the user. The interface to edit settings can be reached via System->Settings if you have the permission to do so.

When editing an item, you can either reset the value to its default value that is given in the view, or change it freely as you wish. The right hand side of the view will give you a preview of the change if you apply them (if possible). Some settings will only accept numbers in a given range and you will not be able to provide wrong input.

1.18 Large exams with 200+ clients

If you want to perform large exams with for example 200+ concurrent clients, there are various settings regarding the webserver and database server that have to be increased. In the following, we assume a goal of (number of concurrent exams you wish to perform) = 200.

1.18.1 Hardware recommendations

See [Hardware Recommendations](#).

1.18.2 GLaDOS

In order for GLaDOS to be able to process 200+ tickets at the same time, you have to increase the setting limiting the maximum number of running daemons under **System->Settings**.

The value n you should choose can be deduced roughly by the formula $n = (\text{number of concurrent exams you wish to perform})/5 + 10$.

1.18.3 System

Increase the upper limit of the total number of inotify instances and watches per user. You can set this on the fly by

```
echo 1024 >/proc/sys/fs/inotify/max_user_instances
echo 8192 >/proc/sys/fs/inotify/max_user_watches
```

Although this will be set back to the default value after the next reboot. To make it permanent, create a new file `/etc/sysctl.d/10-glados.conf` with contents

```
# total number of inotify instances per user
fs.inotify.max_user_instances = 1024
fs.inotify.max_user_watches = 8192
```

The value you should choose for these two settings can be deduced roughly by the formulae:

- $\text{max_user_instances} = (\text{number of concurrent exams you wish to perform}) + 200$
- $\text{max_user_watches} = (\text{number of concurrent exams you wish to perform}) * 10 + 1000$

1.18.4 Apache

The maximum number of concurrent connections to the webserver have to be increased as well. Edit the file `/etc/apache2/mods-enabled/mpm_prefork.conf` and change (or add) the directives

```
<IfModule mpm_prefork_module>
[...]
    MaxRequestWorkers      300
    ServerLimit             300
[...]
</IfModule>
```

The value n you should choose can be deduced roughly by the formula $n = (\text{number of concurrent exams you wish to perform}) + 100$.

To apply the settings, restart the apache2 service

```
service apache2 restart
```

1.18.5 MySQL/ MariaDB

For the database, there is a setting for the maximum number of concurrent connections as well. Create a file `/etc/mysql/mariadb.conf.d/60-glados.cnf` with contents

```
# maximum number of concurrent connections
max_connections      = 350
```

The value `n` you should choose can be deduced roughly by the formula `n = (number of concurrent exams you wish to perform) + (maximum number of running daemons) + 100`. For the maximum number of running daemons, see [System->Settings](#).

To apply the settings, restart the mariaDB service

```
service mariadb restart
```

1.19 LDAP Authentication

This article covers the authentication of users over LDAP. If you plan to use LDAP for authentication on GLaDOS, you have to think about how you want your users to login. LDAP servers provide multiple attributes that are suitable for a login.

To add a new authentication method click [System->Authentication Methods](#) and [Add new Authentication Method](#). In the first step you have to choose an authentication type.

1.19.1 Step 1: Choose an authentication type

There are some LDAP implementations that are directly supported, and therefore need no special setup. If yours is not listed, please choose [Generic LDAP](#) in the dropdown list.

1.19.2 Step 2: Setup

You can define a [Name](#) and [Description](#) of your LDAP server, for example the school name or the domain name.

The [Order](#) field is only needed if you plan to use more than one LDAP server. For more information about the [Order](#) field, please read the article [Multiple LDAP Servers and/or Active Directories](#) first.

Domain / LDAP URI

The configuration option [Domain](#) should be set to the full name of your LDAP domain, for example `example.com`. Your server(s) should be accessible by this DNS name over the network. If this is the case, you don't have set an LDAP URI by hand (and you should not). In that case the LDAP URI is constructed from the LDAP Domain name. Only if your domain name is not resolvable over DNS, you have to provide an LDAP URI explicitly in the [Expert Settings](#).

Login Scheme

For more information about the Login Scheme field, please read the article [Login Scheme](#) first.

Group Mapping

You may want to map some groups defined in your LDAP directory to user roles used in GLaDOS. Members of specific groups may become admins and others may become teachers in GLaDOS. When editing or creating a new authentication method, you can specify the group mapping. If you don't know the group names on your LDAP servers, you can read them out in the form by providing a username to query the LDAP. Click **Query for LDAP groups** in the group mapping configuration option. In the appearing window provide login credentials and proceed with **Query**. If the credentials where valid, the dropdown lists will be populated with group names recieved from the LDAP server(s). You can choose multiple groups to be mapped to the same role in GLaDOS. You may also want to map no group to some roles.

Expert Settings

Most of these settings you usually don't have to change. If you have chosen **Generic LDAP** in **Step 1**, then you may have to provide these settings. They should be available in the documentation of the LDAP implementation you're using.

LDAP URI

Here you can provide a full LDAP URI of the form `ldap://hostname:port` or `ldaps://hostname:port` for SSL encryption. You can also provide multiple LDAP-URIs separated by a space as one string. Note that `hostname:port` is **not** a supported LDAP URI as the schema is missing. See `ldap_connect()` under `ldap_uri`.

Choosing a bind method

The way GLaDOS contacts your LDAP server(s) can be changed. There are 3 methods to choose from in the setup form:

Method 1: Bind directly by the login credentials

If you choose this method, the username provided in the login form is taken directly as bind username to authenticate over LDAP. User details - such as group membership and various needed attributes - are then queried with the login username. Therefore you can only employ this method if the login user has the permission to browse the LDAP directory tree. This is not the default setting in all LDAP implementations, but very common.

The username that should be used for the bind can be modified in a simple way. There are 2 configuration options affecting this: **Bind Scheme** and **Login Scheme**.

The **Bind Scheme** is a pattern to build the bind DN for the actual bind to the LDAP server. `{username}` is replaced with the username extracted from **Login Scheme**.

To understand how the username is extracted, please refer to [Login Scheme](#).

Examples of **Bind Schemes**:

1. `{username}`: no special altering, the extracted username is taken as bind DN as it is.
2. `{username}@foo`: the extracted username is appended with `@foo` to construct the bind DN.

3. `foo\{username}`: the username is prepended with `foo\` for the bind.
4. `{username}@{domain}`: the username is appended with `@{domain}`, where `{domain}` is replaced with the value given in the configuration.
5. `cn={username},dc=example,dc=com`: a distinguished name is built out of the provided username. Instead of `dc=example,dc=com`, one could have also used `{base}`.

Assume your domain name is `example.com` and your LDAP allows to bind with the distinguished Name `dn` as well as with `username@example.com`. To illustrate the power of Bind Scheme together with Login Scheme to rewrite bind credentials, consider the following examples:

where the Example Login column denotes a login username given in the login form and Constructed Bind DN shows the bind DN that is constructed using the login username and Login Scheme together with Bind Scheme. A constructed bind DN of `none` means that there is no bind DN constructed, because the Example Login does not match the Login Scheme, and thus the login has failed.

Method 2: Bind anonymously

The second method is to bind with an anonymous user. To use this method, your LDAP server must allow anonymous binds. When a user attempts to login, the LDAP directory is browsed with an anonymous bind for the Bind Attribute of the login user and his/her group membership. Then GLaDOS performs a second bind, but this time with the value of the users Bind Attribute as bind DN and the provided password. Using this method, you are able to choose the Login Attribute, which is the attribute that should be used for the login.

See the following example configuration:

```
loginAttribute = 'mail';
bindAttribute = 'dn';
```

With that configuration, the user is able to login with his/her E-Mail address deposited in the LDAP directory in the mail attribute (this attribute should be unique across the directory). Most LDAP servers only allow to authenticate with the distinguished name `dn`.

The attribute name of the distinguished name can vary across LDAP implementations. Microsofts Active Directory for example uses `distinguishedName` instead of `dn`.

Method 3: Bind by given username and password

If your LDAP server does not allow anonymous binds, but you anyhow want to use a special attribute as Login Attribute, you can choose this method. For this you have to provide credentials of an account that has the permission to browse the LDAP directory. The advantage of this method is that only that specific user account needs the permission to browse the LDAP directory - the login user itself does not need any permissions, except to bind. However most LDAP implementations do allow every user to browse the LDAP tree. The 2 configuration settings - Login Attribute and Bind Attribute - behave exactly the same as in the anonymous bind scenario above.

1.19.3 Step 3: Test Login

After you created an new authentication method successfully you may want to test login.

Please refer to [Test Login](#) for more information about this.

1.19.4 Step 4: Migrate Users

You may have some local users that now you want to authenticate over the just created authentication method. IN that case you have to perform a user migration.

Please refer to [User Migration](#) for more information about this.

1.20 Active Directory Authentication (Simple)

This article covers the authentication of users over Active Directory. This is a special case of [LDAP Authentication](#).

1.20.1 Step 1: Choose an authentication type

When creating a new authentication method, choose **Microsoft Active Directory** from the dropdown list. In doing so, all special settings are adjusted to Active Directory LDAP.

1.20.2 Step 2: Setup

If your Active Directory servers are reachable by the GLaDOS server and your Active Directory Domain Name can be looked up via DNS, you can provide the Active Directory Domain Name in the **Domain** field and specify a group mapping and you're done!

If you did not change any default settings on your AD, then your users are now able to login to GLaDOS in the same way as they are able to login to a domain joined Windows machine using their **sAMAccountName**.

For the group mapping, you can optionally query for AD groups by pressing **Query for LDAP groups** and providing a username and password. The dropdown lists will then be populated with the query result.

1.20.3 Troubleshooting

Please refer to the [advanced setup](#) if you have trouble configuring Active Directory.

1.21 Active Directory Authentication (Advanced)

This article covers the authentication of users over Active Directory. The advanced setup is very similar to the general LDAP setup, so please first read [LDAP Authentication](#).

1.21.1 Step 1: Choose an authentication type

When creating a new authentication method, choose `Microsoft Active Directory` from the dropdown list. In doing so, all special settings are adjusted to Active Directory LDAP.

1.21.2 Step 2: Setup

Domain / LDAP URI

Please refer to [LDAP Authentication](#) for more information about these fields.

Login Scheme

For more information about the `Login Scheme` field, please read the article [Login Scheme](#) first.

Group Mapping

Please refer to [LDAP Authentication](#) for more information about these fields.

Expert Settings

Choosing a bind method

As bind Method you can choose between `bind directly by the login credentials` and `bind by given username and password`. As for the `bind with anonymous user` method, you have to explicitly allow this on any Active Directory server. So this is not recommended.

Method 1: Bind directly by the login credentials

Active Directory LDAP servers allow the user to bind with both the `distinguishedName` or the `userPrincipalName` attribute. Usually the `userPrincipalName` is the username (`SAMAccountName`) glued with an `@` to the AD domain name.

Assume your AD domain name is `example.com`. Below are a few examples how to set up the authentication method for different outcomes:

Method 2: Bind anonymously

This method will not work unless you have explicitly allowed this in you AD configuration.

Method 3: Bind by given username and password

For this you have to provide credentials of an account that has the permission to browse the AD directory. The advantage of this method is that only that specific user needs permissions to browse the LDAP directory - the login user itself does not need any permissions.

A common setup could look like this:

```
loginAttribute = 'mail';
bindAttribute = 'userPrincipalName';
```

With that configuration, the user is able to login with his/her E-Mail address deposited in the AD directory (this is not default) in the mail attribute.

For Active Directory the Bind Attribute can only be either userPrincipalName or distinguishedName.

1.22 LDAP with SSL

If the Glados server does not trust the certificate chain of a secure LDAP server the [test login](#) form issues an error Can't contact LDAP server or similar.

In order to connect to an LDAP server via SSL, the system running Glados has to trust the secure LDAP server certificate. You may have to include the CA certificate of your LDAP/AD server. First you need to export the CA certificate of your LDAP server in the crt format.

On Debian you can do the following. Copy the CA certificate Domain-CA.crt to /usr/local/share/ca-certificates/

Then run the program that updates the certificate store of your server:

```
update-ca-certificates -v
```

To test whether the certificate is trusted, use:

```
echo | openssl s_client -showcerts -connect hostname.domain.local:636
```

where hostname.domain.local denotes the FQDN of the LDAP server and 636 is the LDAPS port number. You should observe an output like the following if the certificate was trusted:

```
CONNECTED(00000003)
depth=1 DC = local, DC = domain, CN = Domain-CA
verify return:1
depth=0
verify return:1
---
Certificate chain
 0 s:
   i:DC = local, DC = domain, CN = Domain-CA
  -----BEGIN CERTIFICATE-----
  [...]
  -----END CERTIFICATE-----
  ---
Server certificate
```

(continues on next page)

(continued from previous page)

```

subject=

issuer=DC = local, DC = domain, CN = Domain-CA

---
No client certificate CA names sent
Client Certificate Types: RSA sign, DSA sign, ECDSA sign
Requested Signature Algorithms:
  ↳RSA+SHA256:RSA+SHA384:RSA+SHA1:ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA1:DSA+SHA1:RSA+SHA512:ECDSA+SHA512
Shared Requested Signature Algorithms:
  ↳RSA+SHA256:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA384:RSA+SHA512:ECDSA+SHA512
Peer signing digest: SHA256
Peer signature type: RSA
Server Temp Key: ECDH, P-384, 384 bits

---
SSL handshake has read 2057 bytes and written 487 bytes
Verification: OK

---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol   : TLSv1.2
    Cipher     : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: 4525000086F0C2A132B4B44AF39E8AE64AD18F8D1B5ABC64B575B19B50E93045
    Session-ID-ctx:
    Master-Key:
  ↳31E243F284265AE6AC15C9BFF8FAFA8FB5F3F9E0BE8E6989B0D1BC5EFA165DE4BC0F89A61E6416BA1BAE4F866C6018F5
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1632386882
    Timeout    : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: yes

---
DONE

```

If the certificate was not trusted instead it may look like this:

```

CONNECTED(00000003)
depth=0 CN = hostname.domain.local
verify error:num=20:unable to get local issuer certificate
verify return:1
[...]
SSL handshake has read 2115 bytes and written 487 bytes
Verification error: unable to verify the first certificate
[...]
SSL-Session:

```

(continues on next page)

(continued from previous page)

```
[...]
Protocol   : TLSv1.2
Cipher     : ECDHE-RSA-AES256-GCM-SHA384
[...]
Verify return code: 21 (unable to verify the first certificate)
[...]
```

As soon as the verification using the `openssl` command is successful you can [authenticate via LDAP](#) using SSL. For this, choose `ldaps` as Connection Method and 636 as LDAP Port or prefix your LDAP URI with `ldaps://`.

1.23 Test Login

After setting up and authentication method, you should be able to login with user accounts over that authentication method. If the login is not working as desired, you can simulate a login by clicking **Actions->Test Login** in the authentication method view of the desired method or in the authentication method index view. In the dropdown list you can choose to test the login over a specific authentication method. After providing some test login credentials, you will see debug output on the right hand side. With this additional output, you should be able to figure out the problems that occur during login and adjust your setup. The example image below shows some example debug output that may occur in the successful case:

Test Credentials

Method LDAP (Generic LDAP)

Username example.user

Password ••••••••

Test Login

```

Username 'example.user' matches loginScheme {username}.
Opening LDAP connection: ldap://example.com:389.
Setting LDAP_OPT_PROTOCOL_VERSION to 3.
Setting LDAP_OPT_REFERRALS to 0.
Setting LDAP_OPT_NETWORK_TIMEOUT to 5.
Bind with username cn=binduser,dc=example,dc=com successful.
Querying LDAP for the user object with search filter (& (objectClass=posixAccount)
(uid=example.user) ) and base dn dc=example,dc=com for the attributes dn, uidNumber,
gidNumber, uid.
Retrieving 1 entries.
Querying LDAP for group membership with search filter (& (objectClass=posixGroup) (
(memberUid=example.user) (gidNumber=500)) ) and base dn dc=example,dc=com for the attributes
cn.
Retrieving 1 entries.
Bind with username cn=Example User,dc=example,dc=com successful.
User group membership: admins.
User role set to admin.

Authentication was successful.
```

Test

Login Example

1.24 User Migration

This article covers the topic on user migration. User migration is needed if you have changed your authentication method A to another authentication method B and you want your existing users to authenticate over the new method. For example, you created 20 users locally on GLaDOS and now you integrated LDAP authentication and you want these 20 users to authenticate over LDAP from now on. In such a case you have to perform a user migration.

Start the user migration form under **System->Authentication Method** and then click **Actions->Migrate User**.

To be able to migrate users you have to set up an authentication method first, for example [LDAP](#) or [Active Directory](#)

1.24.1 Step 1: Setup

The first thing you have to choose is from which to which method you want to migrate. In the above example you should choose `Local (Database)` in the `From` dropdown list and your new authentication method in the `To` field. Depending on what you have chosen, the next step will be a bit different.

1.24.2 Step 2: Query for users

In this step you have to query for users that are able to migrate. You can do so by pressing the `Query for Users` button. You may have to provide credentials for this step. Depending on your migration setup, the dropdown field `Users to migrate` will be filled with users that are able to be migrated.

Which users are able to migrate depends on your setup.

Essentially, all users associated to the authentication method `From` will be checked whether they could authenticate over the authentication method `To`. In the above example, the local database will be queried for users that authenticate locally. For each found user, the LDAP server is queried whether the user exists in the LDAP directory or not. So each user existing in both authentication methods (`From` and `To`) will be suggested for migration. The dropdown list is then filled with a list of such users. The query can be modified by the `Migrate Search Pattern` to restrict to usernames matching the pattern.

1.24.3 Step 3: Select users to migrate

As a last step, you have to explicitly select the users to want to migrate in the dropdown list. Click `Migrate` to migrate all selected users.

1.24.4 Summary

You will see a summary information about each user migrated of the process after you initiated the user migration. The migrated users can now login over the new authentication method.

Migrated users can always be migrated back to their original authentication method.

1.25 Multiple LDAP Servers and/or Active Directories

This article covers the authentication of users over multiple LDAP domains and/or Active Directory domains. You have the possibility to provide as many authentication methods as you want. You can also combine different LDAP servers with different domains.

Under `System->Authentication Methods` you see a list of all defined authentication methods in their order of processing. One method is always available: the local database. This entry is neither editable nor deletable and the `Login Scheme` is fixed. To deal with multiple authentication methods, you have to specify an `Order` in which they have to be processed during a login. The first item with `Order 0` is the local user database maintained by GLaDOS itself. The item with `Order 0` is reserved for the local user database.

All login attempts will first be authenticated through this local method. This cannot be changed!

If the authentication fails via local database in any way (username does not exist / password is wrong), the authentication method list will be processed further in the given order. When creating a new authentication method it will automatically be appended to the end of the list. The order can be changed by editing the `Order` setting of each involved authentication method.

When dealing with multiple authentication methods, you are encouraged to use different Login Schemes for each method. Please read [Login Scheme](#) before you start to handle multiple authentication methods.

1.25.1 Example 1

Let's have a look at an example. Assume you have two Active Directory domains `domain1.local` and `domain2.local` with Order 1 and 2 respectively. There may be usernames that appear in both ADs. You can set up the two authentication methods as follows:

- The Login Scheme for `domain1.local` set to `{username}@domain1.local`.
- The Login Scheme for `domain2.local` set to `{username}@domain2.local`.

Your users can now login via both domains by login with their traditional usernames appended with `@domain1.local` or `@domain2.local` depending on to which AD they belong to. If a user happens to exist in both domains with exactly the same username, both users can actually log in by just appending their domain to the username. A login of a user existing in `domain1.local` may look like this:

1. User types `user@domain1.local` into the login form
2. Local login attempt (Order 0)
3. If it fails, check Login Scheme of `domain1.local` (Order 1) which matches. Login attempt on `domain1.local` (Order 1)
4. If it fails, check Login Scheme of `domain2.local` (Order 2), but `user@domain1.local` does not match `{username}@domain2.local`
5. Further processing next entry (Order 3)

A login of a user existing in `domain2.local`, would then look like this:

1. User types `user@domain2.local` into the login form
2. Local login attempt (Order 0)
3. If it fails, check Login Scheme of `domain1.local` (Order 1), but `user@domain2.local` does not match `{username}@domain1.local`
4. If it fails, check Login Scheme of `domain2.local` (Order 2) which matches. Login attempt on `domain2.local` (Order 2)
5. Further processing next entry (Order 3)

As you can see, the Login Scheme can be used to make login usernames match a scheme and skip some authentication methods based on that scheme.

1.25.2 Example 2

You can also set up the two authentication methods as follows:

- The Login Scheme for `domain1.local` set to `{username}`.
- The Login Scheme for `domain2.local` set to `{username}@domain2.local`.

In this example all users from the AD whose domain name is `domain1.local`, can log in via their traditional usernames, without and appendix. Users of the domain `domain2.local` although must append the domain to their usernames.

1.25.3 Example 3

You can also set up the two authentication methods as follows:

- The `Login Scheme` for `domain1.local` set to `{username}`.
- The `Login Scheme` for `domain2.local` set to `{username}`.

In this example, all users of both domains can login with their traditional usernames, without appendix. But notice that if a user happens to exist in both domains with exactly the same username, it can no longer be distinguished by GLaDOS. If that user is logging in, the authentication flow is as follows:

1. Local login attempt (Order 0)
2. If it fails, login attempt on `domain1.local` (Order 1), because it matches the `Login Scheme {username}`
3. If it fails, login attempt on `domain2.local` (Order 2), because it matches the `Login Scheme {username}`
4. If it fails, the login has failed.

As you can see, with the `Login Scheme` configuration option you can route the login flow, when dealing with multiple authentication methods.

1.26 Placeholders

Below you will find a list of placeholders that are replaced by the value of the configuration setting or a variable:

1.27 Login Scheme

Every authentication method has a field called `Login Scheme`. This is a pattern to test the given login username against. A login attempt via the associated authentication method will only be performed, if the given username matches the provided pattern. This is used to manage multiple authentication servers and methods, for example multiple LDAP or Active Directory servers with different domains, but can also be used in an environment with one single LDAP server.

`{username}` is extracted from the username provided in the login form according to the `Login Scheme`.

Examples:

- `{username}`: no special testing, all usernames provided are considered for authentication.
- `{username}@foo`: only usernames ending with `@foo` are considered.
- `foo\{username}`: only usernames starting with `foo\` are considered.
- `{username}@{domain}`: only usernames ending with `@{domain}` are considered.

The following table gives some more detailed examples on how the `Login Scheme` affects user authentication:

In the above examples, `{domain}` is a placeholder for the configuration setting `Domain`. You can also use other placeholders. For a full list, please refer to [Placeholders](#).