

---

# GLPI Installation

GLPI Project, Teclib'

2025 年 10 月 13 日



<b>1</b>	<b>要求</b>	<b>3</b>
1.1	Web 服务器	3
1.1.1	Apache configuration	3
1.1.2	Nginx configuration	4
1.1.3	lighttpd configuration	5
1.1.4	IIS configuration	5
1.2	PHP	6
1.2.1	必备扩展依赖	6
1.2.2	可选的扩展	6
1.2.3	Security configuration for sessions	7
1.3	数据库	7
<b>2</b>	<b>安装 GLPI</b>	<b>9</b>
2.1	选择版本	9
2.2	下载	9
2.3	安装	10
2.4	文件和目录位置	10
2.4.1	Plugins files locations	11
<b>3</b>	<b>安装向导</b>	<b>13</b>
3.1	选择语言 (请选择您的语言)	13
3.2	授权	13
3.3	安装 / 升级	14
3.3.1	环境检查	15
3.3.2	连接数据库	16
3.3.3	选择数据库	16
3.3.4	数据库初始化	18
3.3.5	遥测信息	18
3.3.6	安装结束	18
<b>4</b>	<b>Timezones</b>	<b>21</b>
4.1	Non windows users	21
4.2	Windows users	22
4.3	Grant access	22
<b>5</b>	<b>更新</b>	<b>23</b>

<b>6</b>	<b>命令行工具</b>	<b>25</b>
6.1	Console options . . . . .	25
6.2	Additional install and update tools . . . . .	26
6.2.1	Check requirements . . . . .	26
6.2.2	Enable/Disable maintenance . . . . .	26
6.3	安装 . . . . .	26
6.4	Database connection configuration . . . . .	26
6.5	更新 . . . . .	27
6.6	Security key . . . . .	27
6.7	Various tools . . . . .	28
6.7.1	Database schema check . . . . .	28
6.7.2	LDAP synchronization . . . . .	29
6.7.3	Task unlock . . . . .	29
6.8	Suspend/Resume plugins execution . . . . .	30
6.9	Plugins tools . . . . .	30
6.10	Migration tools . . . . .	30
6.10.1	From MyISAM to InnoDB . . . . .	30
6.10.2	Missing timestamps builder . . . . .	30
6.10.3	Use timestamp data type . . . . .	30
6.10.4	Migrate Domains plugin . . . . .	31
6.10.5	Migrate Racks plugin . . . . .	31
<b>7</b>	<b>Advanced configuration</b>	<b>33</b>
7.1	SSL connection to database . . . . .	33

本文为您呈现 **GLPI** 安装介绍。

**GLPI** (即 **Gestion Libre de Parc Informatique** 的首字母简写缩写 **GLPI**) 不花钱 (比如“free speech”非“free beer”!) 资产以及技术支持管理从硬件组件和软件库存管理到用户帮助台管理, 构建的用于管理所有资产管理问题的 web 浏览器都可以访问帮助台管理解决方案。



GLPI 是一款 Web 应用程序它需要:

- web 服务器;
- PHP;
- 一个数据库.

## 1.1 Web 服务器

GLPI 要求 Web 服务器能支持 PHP, 例如:

- Apache 2 (或更新版);
- Nginx;
- lighttpd;
- Microsoft IIS.

### 1.1.1 Apache configuration

Here is a virtual host configuration example for Apache 2 web server.

 **警告**

The following configuration is only suitable for GLPI version 10.0.7 or later.

```
<VirtualHost *:80>
  ServerName glpi.localhost

  DocumentRoot /var/www/glpi/public
```

(续下页)

(接上页)

```
# If you want to place GLPI in a subfolder of your site (e.g. your virtual host is serving
↳multiple applications),
# you can use an Alias directive. If you do this, the DocumentRoot directive MUST NOT
↳target the GLPI directory itself.
# DocumentRoot /var/www/html
# Alias "/glpi" "/var/www/glpi/public"

<Directory /var/www/glpi/public>
    Require all granted

    RewriteEngine On

    # Ensure authorization headers are passed to PHP.
    # Some Apache configurations may filter them and break usage of API, CalDAV, ...
    RewriteCond %{HTTP:Authorization} ^(.+)$
    RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

    # Redirect all requests to GLPI router, unless file exists.
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*)$ index.php [QSA,L]
</Directory>
</VirtualHost>
```

### 备注

If you cannot add specific rewrite rules to the Apache configuration (e.g. you are using a shared hosting), you can use a `.htaccess` file. Still, you need to define the `DocumentRoot` to be the `/public` directory of GLPI.

```
# /var/www/glpi/public/.htaccess
RewriteBase /
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php [QSA,L]
```

## 1.1.2 Nginx configuration

Here is a configuration example for Nginx web server using `php-fpm`.

### 警告

The following configuration is only suitable for GLPI version 10.0.7 or later.

```
server {
    listen 80;
    listen [::]:80;

    server_name glpi.localhost;

    root /var/www/glpi/public;

    location / {
```

(续下页)

(接上页)

```

    try_files $uri /index.php$is_args$args;
}

location ~ ^/index\.php$ {
    # the following line needs to be adapted, as it changes depending on OS distributions_
↔and PHP versions
    fastcgi_pass unix:/run/php/php-fpm.sock;

    fastcgi_split_path_info ^(.+\.(php))(/.*)$;
    include fastcgi_params;

    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
}

```

### 1.1.3 lighttpd configuration

Here is a virtual host configuration example for lighttpd web server.



**警告**

The following configuration is only suitable for GLPI version 10.0.7 or later.

```

$HTTP["host"] =~ "glpi.localhost" {
    server.document-root = "/var/www/glpi/public/"

    url.rewrite-if-not-file = ( "" => "/index.php${url.path}${qsa}" )
}

```

### 1.1.4 IIS configuration

Here is a web.config configuration file example for Microsoft IIS. The physical path of GLPI web site must point to the public directory of GLPI (e.g. D:\glpi\public), and the web.config file must be placed inside this directory.

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Rewrite to GLPI" stopProcessing="true">
          <match url="^(.*)$" />
          <conditions>
            <add input="{REQUEST_FILENAME}" matchType="IsFile" ignoreCase="false"
↔negate="true" />
          </conditions>
          <action type="Rewrite" url="index.php" appendQueryString="true" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>

```

### 警告

The [URL Rewrite](#) module is required.

## 1.2 PHP

表 1: PHP Compatibility Matrix

GLPI version	Minimum PHP version
10.0.X	7.4
11.0.X	8.2

### 备注

GLPI compatibility with new PHP versions is validated shortly after their release. We therefore recommend using the most recent version, for better performances.

### 1.2.1 必备扩展依赖

应用需要以下 PHP 扩展才能正常工作:

- `dom`, `fileinfo`, `filter`, `libxml`, `simplexml`, `tokenizer`, `xmlreader`, `xmlwriter`: these PHP extensions are enable by default and are used for various operations;
- `curl`: used for remote access to resources (inventory agent requests, marketplace, RSS feeds, ...);
- `gd`: used for images handling;
- `intl`: used for internationalization;
- `mysqli`: used for database connection;
- `session`: used for sessions support;
- `zlib`: used for handling of compressed communication with inventory agents, installation of gzip packages from marketplace and PDF generation.

For GLPI 11.0, these additional PHP extensions are mandatory:

- `bcmath`: used for QRcodes generation;
- `mbstring`: used for multibyte chars support and charset conversion;
- `openssl`: used for handling of encrypted communication with inventory agents and OAuth 2.0 authentication.

### 1.2.2 可选的扩展

#### 备注

即使这些扩展不是强制性的，我们建议您无论如何都要安装它们。

对于 GLPI 的一些额外特性，需要以下 PHP 扩展:

- `bz2`, `Phar`, `zip`: enable support of most common packages formats in marketplace;

- `exif`: enhance security on images validation;
- `ldap`: enable usage of authentication through remote LDAP server;
- `openssl`: enable email sending using SSL/TLS;
- `Zend OPcache`: enhance PHP engine performances.

Following PHP extensions are emulated in GLPI and can be enabled to slightly improve performances: `* ctype`; `* iconv`; `* sodium`.

### 1.2.3 Security configuration for sessions

To enhance security, it is recommended to configure PHP sessions with the following settings:

- `session.cookie_secure`: should be set to `on` when GLPI can be accessed on **only** HTTPS protocol;
- `session.cookie_httponly`: should be set to `on` to prevent client-side scripts from accessing cookie values;
- `session.cookie_samesite`: should be set, at least, to `Lax`, to prevent cookies from being sent cross-origin (across domains) POST requests.

#### 备注

Refer to [PHP documentation](#) for more information about session configuration.

## 1.3 数据库

为了工作，GLPI 需要一个数据库服务器。

#### 警告

Currently, only [MySQL](#) and [MariaDB](#) database servers are supported by GLPI.

表 2: Database server compatibility matrix

GLPI version	Database server	Minimum version
10.0.X	MySQL	5.7
10.0.X	MariaDB	10.2
11.0.X	MySQL	8.0
11.0.X	MariaDB	10.6

#### 备注

GLPI compatibility with new LTS database servers versions is validated shortly after their release. We therefore recommend using a recent version, for better performances.

**ICC-by-NC-NDI**



请按以下步骤操作：

1. *Configure your webserver*,
2. 版本选择,
3. 下载档案,
4. 安装:)

## 2.1 选择版本

### 📌 备注

强烈建议您为生产用途选择最新的稳定版本。

GLPI 遵循 3 位数的语义版本控制方案。第一个是主要版本，第二个是次要版本，第三个是修订版本。主要版本可能具有重要的不兼容性和新功能；次要版本也可能带来新功能，但在主要版本内保持完美兼容。修复版本只会修复报告的问题而不添加任何新内容。

## 2.2 下载

### ⚠️ 警告

在 GitHub 上，总有两个名为 *\*Source code\** 的档案，不应该使用。不应该这样说啊，github 上不使用的 source code 好像没这说法。

点击 下载按钮 GLPI 官方网站 (或从 Github 获取 [Github release](#)) 选择 `glpi-{version}.tgz` 文档。

### 2.3 安装

GLPI 安装由三个步骤组成：

1. 解压缩您网站中的存档；
2. 让你的 web 服务器写入“files”和“config”目录；
3. Launch *installation wizard* (or use the *command line installation script*).

完成这三个步骤后，即可使用该应用程序。

If you need to set advanced configuration, like SSL database connection parameters, please refer to *advanced configuration*.

### 2.4 文件和目录位置

As long as the web server is configured properly (see *web server configuration*), sensible files should not be exposed publicly.

GLPI stores some data in the `files` directory, the database access configuration is stored in the `config` directory, etc. To ease the GLPI maintenance, the location of GLPI storage directories can be customized.

There are a few configuration directives you may use to achieve that:

- `GLPI_CONFIG_DIR`: 设置配置目录的路径；
- `GLPI_VAR_DIR`: 设置 `files` 目录路径；
- `GLPI_LOG_DIR`: 设置日志文件路径。

#### **i** 备注

There are many other configuration directives available, the ones we talked about are the main to move all stored files outside the GLPI source code.

目录的选择完全取决于你；以下示例将遵循‘FHS <<http://www.pathname.com/fhs/>>’建议。

我们的 GLPI 实例将安装在“/var/www/glpi”中，Web 服务器配置中的特定虚拟主机将反映此路径。

GLPI configuration will be stored in `/etc/glpi`, just copy the contents of the `config` directory to this place. GLPI requires read rights on this directory to work; and write rights during the installation process.

GLPI data will be stored in `/var/lib/glpi/files`, just copy the contents of the `files` directory to this place. GLPI requires read and write rights on this directory.

GLPI 日志文件将存储在“/var/log/glpi”中，此处无需复制，只需创建目录即可。GLPI 需要对此目录进行读写访问。

按照这些说明，我们将在 GLPI 目录中创建一个“inc/downstream.php”文件，其中包含以下内容：

```
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');

if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
    require_once GLPI_CONFIG_DIR . '/local_define.php';
}
```

**警告**

GLPI 包肯定会提供一个“inc/downstream.php”文件。这个不能编辑！

GLPI looks for a *local\_define.php* file in its own *config* directory. If you want to use one from new config directory, you have to load it.

然后，使用以下内容在“/etc/glpi/local\_define.php”中创建一个文件：

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi/files');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

**备注**

GLPI\_VAR\_DIR permits to change the storage path of all the GLPI files, but you can adapt the storage path for each kind of files.

```
<?php
define('GLPI_VAR_DIR',          '/var/lib/glpi/files');
define('GLPI_DOC_DIR',         GLPI_VAR_DIR);           // Path for documents storage
define('GLPI_CACHE_DIR',      GLPI_VAR_DIR . '/_cache'); // Path for cache storage
define('GLPI_CRON_DIR',       GLPI_VAR_DIR . '/_cron');  // Path for cron storage
define('GLPI_GRAPH_DIR',     GLPI_VAR_DIR . '/_graphs'); // Path for graph storage
define('GLPI_LOCAL_I18N_DIR', GLPI_VAR_DIR . '/_locales'); // Path for local i18n files
define('GLPI_LOCK_DIR',      GLPI_VAR_DIR . '/_lock');  // Path for lock files storage.
↳ (used by cron)
define('GLPI_LOG_DIR',        GLPI_VAR_DIR . '/_log');   // Path for log storage
define('GLPI_PICTURE_DIR',    GLPI_VAR_DIR . '/_pictures'); // Path for picture storage
define('GLPI_PLUGIN_DOC_DIR', GLPI_VAR_DIR . '/_plugins'); // Path for plugins documents.
↳ storage
define('GLPI_RSS_DIR',        GLPI_VAR_DIR . '/_rss');   // Path for RSS feeds storage
define('GLPI_SESSION_DIR',    GLPI_VAR_DIR . '/_sessions'); // Path for sessions files.
↳ storage
define('GLPI_TMP_DIR',        GLPI_VAR_DIR . '/_tmp');   // Path for temporary files.
↳ storage
define('GLPI_UPLOAD_DIR',     GLPI_VAR_DIR . '/_uploads'); // Path for upload storage
define('GLPI_INVENTORY_DIR',  GLPI_VAR_DIR . '/_inventories'); // Path for inventory files.
↳ storage
define('GLPI_THEMES_DIR',     GLPI_VAR_DIR . '/_themes'); // Path for custom themes.
↳ storage
```

## 2.4.1 Plugins files locations

Added in version 11.0.0.

Plugins files location can be configured using the GLPI\_MARKETPLACE\_DIR configuration directive.

To store the plugins in the /var/lib/glpi/plugins directory, just copy the contents of the marketplace and plugins directories to this place. GLPI requires read and write rights on this directory.

Then, in the /etc/glpi/local\_define.php file, add the following contents:

```
define('GLPI_MARKETPLACE_DIR', '/var/lib/glpi/plugins');
```

**ICC-by-NC-NDI**



正式开始安装, 浏览器地址栏敲入 GLPI 地址: `https://\{adresse_glpi\}/`  
如 GLPI 尚未安装过, 则您下一步下一步即可安装好的。

### 3.1 选择语言 (请选择您的语言)

第一步将允许您选择安装语言。选择您的 lang 语言, 并单击 validate 生效。



### 3.2 授权

GLPI 的使用需要 GNU 许可证的批准。一旦许可条款被阅读和接受, 只需验证表单。



如果您不同意许可条款，则不可能继续安装过程。

### 3.3 安装 / 升级

本页面将允许在新的 GLPI 安装和更新之间进行选择。



点击安装.

### 3.3.1 环境检查

This step will check if prerequisites are met. If they're not, it is not possible to continue and an explicit error message will tell you about what is wrong and what to do before trying again.

**GLPI SETUP**

Step 0

Checking of the compatibility of your environment with the execution of GLPI

Test done	Results
Testing PHP Parser	✓
Sessions test	✓
Test if Session_use_trans_sid is used	✓
mysqli extension test	✓
ctype extension test	✓
fileinfo extension test	✓
json extension test	✓
mbstring extension test	✓
zlib extension test	✓
curl extension test	✓
gd extension test	✓
simplexml extension test	✓
xml extension test	✓
imap extension test	✓
APCu extension test	✓
xmlrpc extension test	✓
ldap extension test	! ldap extension is not present
Zend OPcache extension test	! Zend OPcache extension is not present
Allocated memory test	✓
Checking write permissions for setting files	✓
Checking write permissions for document files	✓
Checking write permissions for dump files	✓
Checking write permissions for session files	✓
Checking write permissions for automatic actions files	✓
Checking write permissions for graphic files	✓
Checking write permissions for lock files	✓
Checking write permissions for plugins document files	✓
Checking write permissions for temporary files	✓
Checking write permissions for cache files	✓
Checking write permissions for rss files	✓
Checking write permissions for upload files	✓
Checking write permissions for pictures files	✓
Checking write permissions for log files	✓
SELinux mode is Enforcing	✓
SELinux boolean configuration for httpd_can_network_connect --> on	✓
SELinux boolean configuration for httpd_can_network_connect_db --> on	✓
SELinux boolean configuration for httpd_can_sendmail --> on	✓

Do you want to continue?

Continue Try again

Some prerequisites are optional, it will be possible to continue installation event if they're not met.

### 3.3.2 连接数据库

询问数据库连接参数。



The screenshot shows the GLPI installation setup interface. At the top left is the GLPI logo. The main heading is 'GLPI SETUP' in yellow. Below it, 'Step 1' and 'Database connection setup' are displayed. A white box contains the 'Database connection parameters' section with three input fields: 'SQL server (MariaDB or MySQL)', 'SQL user', and 'SQL password'. A yellow 'Continue' button is positioned at the bottom center of the form area.

- *MySQL server*: enter the path to your MySQL server, *localhost* or *mysql.domaine.tld* as example;
- *MySQL user*: enter user name that is allowed to connect to the Database;
- *MySQL password*: enter user's password.

正确填写所有字段后，验证格式。

然后建立第一个数据库连接。如果参数无效，将显示错误消息，您必须修复参数并重试。

### 3.3.3 选择数据库

与数据库服务器连接正常后，您必须为 GLPI 创建或选择所需的数据库并进行初始化。



有两种方法：

- 使用已有数据库

在显示的列表中选择此数据库。验证使用。

**警告**

选定的数据库内容将在安装时销毁。

- 新建数据库

选择 \* 创建新数据库 \*，在相关字段中输入数据库名称，然后验证以创建基础。

**警告**

SQL 用户必须能够为此选项创建新数据库才能工作。

### 3.3.4 数据库初始化

此步骤使用默认值初始化数据库。



如果有任何错误；注意显示的信息。

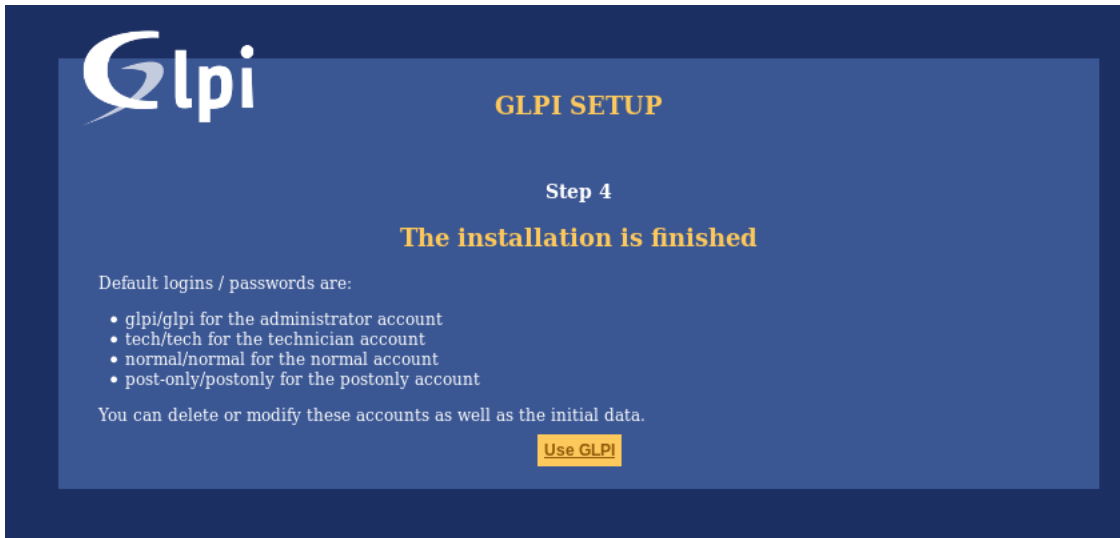
### 3.3.5 遥测信息

GLPI 将要求您分享一些遥测信息并进行注册。这不是强制性的。



### 3.3.6 安装结束

此步骤提供安装摘要并提供所创建的用户列表。请记住此重要信息其将在登陆时需要用到。



#### 备注

默认用户帐号是：

- *glpi/glpi* 为 admin 管理员账号,
- *tech/tech* 技术员账号,
- *normal/normal* ”普通”账号,
- *post-only/postonly* 作业员账号.

#### 警告

出于明显的安全问题，您必须删除或编辑这些帐户。

在删除“glpi”帐户之前，请确保您使用“super-admin”配置文件创建了另一个用户。

**ICC-by-NC-NDI**



In order to get timezones working on a MariaDB/MySQL instance, you will have to initialize Timezones data and grant GLPI database user read ACL on their table.

 警告

Enabling timezone support on your MySQL instance may affect other database in the same instance; be carefull!

 警告

Currently, MySQL, and MariaDB (prior to 11.5), have a maximum date limited to 2038-01-19 on fields relying on timestamp type!

MariaDB 11.5 onwards is limited to 2106.

## 4.1 Non windows users

On most systems, you'll have to initialize timezones data from your system's timezones:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -p -u root mysql
```

You may want to check [MariaDB documentation about mysql\\_tzinfo\\_to\\_sql](#) and your system documentation to know where data are stored (if not in `/usr/share/zoneinfo`).

Do not forget to restart the database server once command is successfull.

### 4.2 Windows users

Windows does not provide timezones informations, you'll have to download and initialize data yourself.

See [MariaDB documentation about timezones](#).

### 4.3 Grant access

Removed in version 10.0.20: Granting access to the `mysql.time_zone_name` is no longer required to support timezones usage.



Be carefull not to give your GLPI database user too large access. System tables should **never** grant access to app users.

In order to list possible timezones, your GLPI database user must have read access on `mysql.time_zone_name` table. Assuming your user is `glpi@localhost`, you should run something like:

```
GRANT SELECT ON `mysql`.`time_zone_name` TO 'glpi'@'localhost';
```



**i 备注**

每次更新，正式执行更新前，请务必注意备份备份备份！

- 备份数据库备份数据库备份数据库;
- backup your `config` directory, especially for your GLPI key files (`config/glpi.key`, `config/glpicrypt.key`, `config/oauth.pem`, `config/oauth.pub`) which are randomly generated;
- backup your `files` directory, it contains users and plugins generated files, like uploaded documents;
- backup your `marketplace` and `plugins` directories.

Here are the steps to update GLPI:

- Download latest GLPI version.
- Ensure the target directory is empty and extract files there.
- Restore the previously backed up `config`, `files`, `marketplace` and `plugins` directories.
- Then open the GLPI instance URI in your browser, or (recommended) use the `php bin/console db:update` *command line tool*.

**⚠ 警告**

As soon as a new version of GLPI files is detected, you will not be able to use the application until the update process has been done.

**⚠ 警告**

您不应该尝试在非空数据库上恢复数据库备份 (例如，由于任何原因部分迁移的数据库)。

在恢复备份并尝试更新之前，请确保数据库为空，并在失败时重复此操作。

### 备注

Before GLPI 11.0, all the plugins were disabled during the update process.

Since GLPI 11.0, as soon as a new version of GLPI files is detected, the plugins execution is suspended. Their execution will be resumed after the update only in case of a minor/bugfixes update. Otherwise, you will have to do it manually using the `php bin/console plugin:resume_execution` command line tool or from the plugins administration page.

### 备注

You can use the `php bin/console db:check` *command line tool* before executing the update command. This will allow you to check the integrity of your database, and to identify changes to your database that could compromise the update.

**ICC-by-NC-NDI**

在 GLPI 9.2.2 版本之前，命令行工具属于非官方但受支持的脚本来提供其可在“scripts”目录中找到。因不属于官方，故不再发布归档。

Since GLPI 9.4.0, command line tools are being centralized in a console application (`bin/console`). Calling `php bin/console` from GLPI directory displays the list of available commands.

#### 备注

假如您系统中已安装了 APCu 依赖，由于在命令行下默认配置其为禁用状态，因此它可能会在命令行模式下失效。如需要启用，请找到 APCu 的配置文件中的 `apc.enable_cli` 设为 `on` 启用。

#### 警告

When using cli tools, please check the system user you are currently logged in with, and permissions on files and directories. With a wrong user, logs, cache and other files may be created with rights that would not allow your webserver to read or write on those files!

## 6.1 Console options

For every console command, following options are available:

- `--config-dir=CONFIG-DIR` path of configuration directory to use, relative to current working directory (required only if a custom path is used)
- `-h, --help` displays command help
- `--lang=LANG` output language code (default value is existing GLPI "language" configuration or "en\_GB")
- `-n, --no-interaction` disable command interactive questions
- `--no-plugins` disable GLPI plugins during command execution

- `-q, --quiet` disable command output
- `-v|vv|vvv, --verbose=VERBOSE` verbosity level: 1 for normal output, 2 for more verbose output and 3 for debug

## 6.2 Additional install and update tools

### 6.2.1 Check requirements

Before installing or upgrading, requirements are automatically checked; but you can run them separately and see state for all of them using the `php bin/console glpi:system:check_requirements` command.

### 6.2.2 Enable/Disable maintenance

GLPI provides a maintenance mode that can be activated prior to an update, and deactivated after all has been checked. Just use the `glpi:maintenance:enable` and `glpi:maintenance:disable` commands.

## 6.3 安装

The `php bin/console db:install` has been made to install GLPI database in CLI mode.

本命令的可能选项是:

- `-r, --reconfigure` to enable overriding of any existing DB configuration file
- `-f, --force` to force execution of installation even if database is not empty
- `-L, --default-language=DEFAULT_LANGUAGE` default language of GLPI (*en\_GB* per default)
- `-H, --db-host=DB_HOST` host name (*localhost* per default)
- `-P, --db-port=DB_PORT` database port (default MySQL port if option is not defined)
- `-d, --db-name=DB_NAME` database name
- `-u, --db-user=DB_USER` database user name
- `-p, --db-password=DB_PASSWORD` database user's password (use it without value to be prompted for password)

If mandatory options are not specified in the command call, the console will ask for them.

Database connection parameters may be omitted if a configuration file already exists.

See also *console options*.

## 6.4 Database connection configuration

Added in version 9.5.0.

The `php bin/console db:configure` has been made to define database connection parameters in CLI mode.

本命令的可能选项是:

- `-r, --reconfigure` to enable overriding of any existing DB configuration file
- `-H, --db-host=DB_HOST` host name (*localhost* per default)
- `-P, --db-port=DB_PORT` database port (default MySQL port if option is not defined)
- `-d, --db-name=DB_NAME` database name

- `-u, --db-user=DB_USER` database user name
- `-p, --db-password=DB_PASSWORD` database user's password (use it without value to be prompted for password)

If mandatory options are not specified in the command call, the console will ask for them.

See also *console options*.

## 6.5 更新

The `php bin/console db:update` has been made to update GLPI database in CLI mode from a previously installed version.

There is no required arguments, just run the command so it updates your database automatically.

### 警告

在进行任何更新尝试之前，别忘记备份数据库备份数据库备份数据库！

### 警告

Since GLPI 10.0.2, `db:check_schema_integrity` is executed before performing the update. If an error is detected, the command will ask you if you want to continue (unless `--no-interaction` is used). You can bypass this `db:check_schema_integrity` by using the option `-s, --skip-db-checks`.

本命令的可能选项是：

- `-u, --allow-unstable` allow update to an unstable version (use it with cautions)
- `-f, --force` force execution of update from v-1 version of GLPI even if schema did not changed
- `-s, --skip-db-checks` do not check database schema integrity before performing the update
- `--enable-telemetry` allow usage statistics sending to Telemetry service (<https://telemetry.glpi-project.org>)
- `--no-telemetry` disallow usage statistics sending to Telemetry service (<https://telemetry.glpi-project.org>)

See also *console options*.

## 6.6 Security key

Added in version 9.4.6.

### 备注

GLPI key file is available for GLPI  $\geq$  9.4.6 but is not mandatory. As of GLPI 9.5, using the key file will be mandatory.

In order to store some sensitive data, GLPI relies on a homemade encryption/decryption tool, which uses a key to:

- encrypt data before storing them in the database,
- decrypt data that has been retrieved from the database.

The `php bin/console glpi:security:change_key` command allows to change the key, if it has been compromised for example. By default, command will:

- generate a new key and store it in the key file,
- update all configured fields (for core and compatible plugins) to use the new key,
- update all configuration entries listed (for core and compatible plugins) to use the new key.

## 6.7 Various tools

### 6.7.1 Database schema check

The `php bin/console db:check_schema_integrity` command can be used to check if your database schema differs from expected one.

本命令的可能选项是:

- `--strict`: Strict comparison of definitions
- `--check-all-migrations`: Check tokens related to all databases migrations.
- `--check-innodb-migration`: Check tokens related to migration from "MyISAM" to "InnoDB".
- `--check-timestamps-migration`: Check tokens related to migration from "datetime" to "timestamp".
- `--check-utf8mb4-migration`: Check tokens related to migration from "utf8" to "utf8mb4".
- `--check-dynamic-row-format-migration`: Check tokens related to "DYNAMIC" row format migration.
- `--check-unsigned-keys-migration`: Check tokens related to migration from signed to unsigned integers in primary/foreign keys.
- `-p, --plugin`: Plugin to check. If option is not used, checks will be done on GLPI core database tables.

If you have any diff, output will look like :

```
$ php bin/console glpi:database:check_schema_integrity
Table schema differs for table "glpi_rulecriterias".
--- Original
+++ New
@@ @@
 create table `glpi_rulecriterias` (
   `id` int(11) not null auto_increment
   `rules_id` int(11) not null default '0'
   `criteria` varchar(255) default null
   `condition` int(11) not null default '0'
- `pattern` text default null
+ `pattern` text
 primary key (`id`)
```

Compared to the GLPI installation file:

- a line that starts with `-` means that something is missing in your database
- a line that starts with `+` means that there is something extra in your database

You can also have a message like `Unknown table "glpi_tablename" has been found in database., this indicates that this table doesn't exist in the installation file of the current GLPI schema:`

- either it's a table that you have voluntarily created for your needs, you can ignore this message
- either it's an old GLPI table which is no longer useful, you can delete it (taking care to make a backup before)

## 6.7.2 LDAP synchronization

The `bin/console glpi:ldap:synchronize_users` command can be used to synchronize users against LDAP server informations.

本命令的可能选项是:

- `-c, --only-create-new` only create new users
- `-u, --only-update-existing` only update existing users
- `-s, --ldap-server-id[=LDAP-SERVER-ID]` synchronize only users attached to this LDAP server (multiple values allowed)
- `-f, --ldap-filter[=LDAP-FILTER]` filter to apply on LDAP search
- `--begin-date[=BEGIN-DATE]` begin date to apply in "modifyTimestamp" filter
- `--end-date[=END-DATE]` end date to apply in "modifyTimestamp" filter
- `-d, --deleted-user-strategy[=DELETED-USER-STRATEGY]` force strategy used for deleted users:
  - 0: Preserve
  - 1: Put in trashbin
  - 2: Withdraw dynamic authorizations and groups
  - 3: Disable
  - 4: Disable + Withdraw dynamic authorizations and groups

See <http://php.net/manual/en/datetime.formats.php> for supported date formats in `--begin-date` and `--end-date` options.

See also *console options*.

## 6.7.3 Task unlock

The `php bin/console task:unlock` command can be used to unlock stucked cron tasks.



**警告**

Keep in mind that no task should be stucked except in case of a bug or a system failure (database failure during cron execution for example).

本命令的可能选项是:

- `-a, --all` unlock all tasks
- `-c, --cycle[=CYCLE]` execution time (in cycles) from which the task is considered as stuck (delay = task frequency \* cycle)
- `-d, --delay[=DELAY]` execution time (in seconds) from which the task is considered as stuck (default: 1800)
- `-t, --task[=TASK]` itemtype::name of task to unlock (e.g: MailCollector::mailgate)

See also *console options*.

## 6.8 Suspend/Resume plugins execution

Added in version 11.0.

The `php bin/console plugin:suspend_execution` and the `php bin/console plugin:resume_execution` commands can be used to suspend or resume the plugins execution.

## 6.9 Plugins tools

Added in version 9.5.

Some command line tools are also available to manage plugins from command line:

- `glpi:plugin:install`
- `glpi:plugin:activate`
- `glpi:plugin:deactivate`

In order to install `MyGreatPlugin`; you should end with something like:

```
$ ./bin/console glpi:plugin:install MyGreatPlugin
$ ./bin/console glpi:plugin:activate MyGreatPlugin
```

Each of those plugin commands can take a plugin name as argument, or the `--all` flag to be ran on all plugins.

## 6.10 Migration tools

### 6.10.1 From MyISAM to InnoDB

Added in version 9.3.0.

Since version 9.3.0, GLPI uses the InnoDB engine instead of previously used MyISAM engine.

The `php bin/console glpi:migration:myisam_to_innodb` command can be used to migrate existing tables to InnoDB engine.

### 6.10.2 Missing timestamps builder

Added in version 9.1.0.

Prior to GLPI 9.1.0, fields corresponding to creation and modification dates were not existing.

The `php bin/console glpi:migration:build_missing_timestamps` command can be used to rebuild missing values using available logs.

### 6.10.3 Use timestamp data type

Added in version 9.5.0.

Many date fields were using the `DATETIME` type, but this does not allow to rely on timezones. Timezone support requires all fields to use `TIMESTAMP` data type, but this query can be very long and therefore is not included in the standard update process.

Using the `glpi:migration:timestamps` command will change those fields to the correct data type, but read *documentation on timezones* before.



Ensure to backup your database before!

## 6.10.4 Migrate Domains plugin

Added in version 9.5.0.

Domains in GLPI have evolved from a simple dropdown to a more complex object, including records management among others. Therefore, the Domains plugins feature are now included in core.

To migrate your plugin data; use the `glpi:migration:domains_plugin_to_core` command. Presence of the plugin is mandatory so checks can be run, you can use the `--without-plugin` switch but this is not recommended. If you were using an older version of the plugin than the one required, you can use the `--update-plugin` flag.

At the end, all domains types, domains and item relations will be migrated in core tables.

## 6.10.5 Migrate Racks plugin

Added in version 9.5.0.

Since GLPI 9.3.0, data center infrastructure management is available as a core feature. A migration script from Racks plugin was provided inside the `scripts` directory. Since GLPI 9.5.0, this migration script has been refactored and moved inside the CLI console.

To migrate your plugin data; use the `glpi:migration:racks_plugin_to_core` command. Presence of the plugin is mandatory so checks can be run, you can use the `--without-plugin` switch but this is not recommended. If you were using an older version of the plugin than the one required, you can use the `--update-plugin` flag.

**ICC-by-NC-NDI**



## 7.1 SSL connection to database

Added in version 9.5.0.

Once installation is done, you can update the `config/config_db.php` to define SSL connection parameters. Available parameters corresponds to parameters used by `mysqli::ssl_set()`:

- `$dbssl` defines if connection should use SSL (*false* per default)
- `$dbsslkey` path name to the key file (*null* per default)
- `$dbsslcert` path name to the certificate file (*null* per default)
- `$dbsslca` path name to the certificate authority file (*null* per default)
- `$dbsslcapath` pathname to a directory that contains trusted SSL CA certificates in PEM format (*null* per default)
- `$dbsslcipher` list of allowable ciphers to use for SSL encryption (*null* per default)

 警告

For now it is not possible to define SSL connection parameters prior or during the installation process. It has to be done once installation has been done.



ICC-by-NC-NDI