

---

# GLPI plugins Documentation

*Version 0.1*

**Teclib'**

**août 04, 2021**



---

## Table des matières

---

<b>1</b>	<b>Présentation</b>	<b>3</b>
<b>2</b>	<b>Empty</b>	<b>5</b>
2.1	Créer un nouveau plugin . . . . .	5
2.2	Mise à jour d'un plugin existant . . . . .	5
2.3	Fonctionnalités . . . . .	6
<b>3</b>	<b>Exemple</b>	<b>11</b>
<b>4</b>	<b>Treeview</b>	<b>13</b>
4.1	Requirements for latest version . . . . .	13
4.2	Features . . . . .	13
4.3	Install the Plugin . . . . .	13
4.4	Usage . . . . .	13
<b>5</b>	<b>Tag</b>	<b>15</b>
5.1	Requirements for latest version . . . . .	15
5.2	Features . . . . .	15
5.3	Install the Plugin . . . . .	15
5.4	Usage . . . . .	16
<b>6</b>	<b>News</b>	<b>19</b>
6.1	Requirements for latest version . . . . .	19
6.2	Features . . . . .	19
6.3	Install the Plugin . . . . .	19
6.4	Usage . . . . .	20
6.5	Create alerts . . . . .	20
6.6	Targets . . . . .	20
6.7	Display on login page . . . . .	21
6.8	Display on helpdesk page . . . . .	21
<b>7</b>	<b>SCCM</b>	<b>23</b>
7.1	Requirements for latest version . . . . .	23
7.2	Features . . . . .	23
7.3	Workflow . . . . .	23
7.4	Schematic diagram . . . . .	24
7.5	Collected data . . . . .	24

7.6	Automatic actions . . . . .	25
<b>8</b>	<b>Escalade</b>	<b>27</b>
8.1	Requirements for latest version . . . . .	27
8.2	Features . . . . .	27
8.3	Install the Plugin . . . . .	27
8.4	Usage . . . . .	27
8.5	Configuration . . . . .	28
8.6	Ticket's search engine . . . . .	30
8.7	Central page . . . . .	31
<b>9</b>	<b>Credit</b>	<b>33</b>
9.1	Requirements for latest version . . . . .	33
9.2	Features . . . . .	33
9.3	Configuration . . . . .	33
9.4	Credit consumption . . . . .	34
9.5	Credit report . . . . .	34
<b>10</b>	<b>Generic Object</b>	<b>37</b>
10.1	Pré-requis . . . . .	37
10.2	Fonctionnalités . . . . .	37
10.3	Exemple d'utilisation . . . . .	37
10.4	Installer le plugin . . . . .	38
10.5	Utilisation . . . . .	38
10.6	Ajouter des champs globaux . . . . .	43
10.7	Configurer les droits . . . . .	43
10.8	Utiliser le nouveau champ . . . . .	43
10.9	Exemple d'utilisation de Generic Object comme GMAO . . . . .	43
<b>11</b>	<b>Order</b>	<b>47</b>
11.1	Requirements . . . . .	47
11.2	Features . . . . .	47
11.3	Install the Plugin . . . . .	47
11.4	Configuration . . . . .	48
11.5	Creation of a reference catalogue . . . . .	49
11.6	Creating an order . . . . .	50
11.7	Rights management . . . . .	55
<b>12</b>	<b>Champs additionnels</b>	<b>57</b>
12.1	Pré-requis . . . . .	57
12.2	Fonctionnalités . . . . .	57
12.3	Installer le plugin . . . . .	57
12.4	Utilisation . . . . .	58
12.5	Recherche . . . . .	61
12.6	Interface simplifiée . . . . .	61
12.7	Traductions . . . . .	61
12.8	Entités . . . . .	62
<b>13</b>	<b>Uninstall</b>	<b>65</b>
13.1	Requirements for latest version . . . . .	65
13.2	Features . . . . .	65
13.3	Right managment . . . . .	65
13.4	Uninstall feature . . . . .	66
13.5	Uninstall hardware . . . . .	67
13.6	Replace feature . . . . .	67

13.7	Replace hardware	69
<b>14</b>	<b>Form creator</b>	<b>71</b>
14.1	Features	71
14.2	Install the Plugin	71
14.3	Service catalog	72
14.4	Forms configuration	73
14.5	Targets	88
14.6	Exporting and importing forms	92
14.7	Contributing	95
14.8	Team	95
<b>15</b>	<b>Datainjection</b>	<b>99</b>
15.1	Requirements for latest version	99
15.2	Features	99
15.3	Install the Plugin	99
15.4	Configuration	100
15.5	Create new model	100
15.6	Inject your CSV file	101
15.7	Mapping CSV column and object field	101
15.8	Additional data	102
15.9	Validate model	102
15.10	Execute import	103
15.11	Import result	104
<b>16</b>	<b>Oauth IMAP</b>	<b>105</b>
16.1	Fonctionnalités	105
16.2	Installer le plugin	105
16.3	Configurer votre Application Oauth IMAP	105
16.4	Créer une autorisation	106
16.5	Configurer votre collecteur mail	106





Vous trouverez ici la documentation de divers plugins GLPI.





# CHAPITRE 1

---

## Présentation

---

Les plugins fournissent la possibilité d'étendre les possibilités de [GLPI](#) sans en modifier le code source. Dans un monde parfait, les plugins devraient ne fournir que des fonctionnalités spécifiques, tandis que le cœur fournit les fonctionnalités que tout le monde (ou presque) trouvera utiles.

Certains plugins existants sont très complexes, d'autres très simples. Certains vont requérir la lecture de leur documentation avant de pouvoir les utiliser, d'autres pas... Certains proposeront des fonctionnalités de configuration, de gestion des droits, entre autres... Ou pas ! Il y a déjà tellement de plugins !

Si vous recherchez des fonctionnalités qui n'existent pas dans le cœur de GLPI, commencez par chercher dans le [dépôt des plugins](#). Vous pourrez chercher un nom de plugin ou une fonctionnalité, voir quelle version du cœur est compatible, etc.

Le but de la présente documentation est de centraliser les documentations, mais nous ne pouvons prétendre que les documentations de tous les plugins soient hébergées ici.

Dans tous les cas, joyeuse utilisation de GLPI !





- Lien des sources : <https://github.com/pluginsGLPI/empty>
- Téléchargement : *non pertinent*

Ce plugin fournit... Rien. Vraiment !

Il est prévu pour être un point de départ pour l'écriture de plugins, avec des versions minimales de scripts utilitaires, et quelques conseils. Si vous recherchez plutôt les possibilités offertes par les plugins, regardez *le plugin exemple*.

### 2.1 Créer un nouveau plugin

Un script utilitaire pour créer un nouveau plugin - `plugin.sh` est fourni. Vous devrez l'appeler avec un nom de plugin, une version et optionnellement le chemin dans lequel vous souhaitez le créer.

```
$ ./plugin.sh MyGreatPlugin 0.0.1
```

Le script nettoiera et passera en minuscules le nom que vous avez fourni, copiera les fichiers modèles dans le nouveau répertoire, et effectuera quelques remplacements.

Si vous utilisez le script sans spécifier de dossier de destination, votre répertoire `mygreatplugin` sera placé tout à côté de son propre dossier. Autrement, le répertoire sera créé dans le chemin spécifié :

```
$ ./plugin.sh MyGreatPlugin 0.0.1 /path/to/glpi/plugins/
```

### 2.2 Mise à jour d'un plugin existant

Il n'y a pas de méthode automatique pour mettre à jour un plugin existant, il y aurait en effet bien trop de cas de figure à gérer. Mais ne vous inquiétez pas, la procédure est assez simple ;)

L'utilisation des possibilités de empty est aussi simple que la création de quelques fichiers :

- `composer.json`,
- `.travis.yml`,

- Robofile.php,
- .gitignore.

Si vous n'avez pas encore de configuration `composer` ou `travis`, vous pouvez simplement copier celles du plugin `empty`. Dans le cas contraire ; ajoutez dans votre `composer.json` :

```
{
  "minimum-stability": "dev",
  "prefer-stable": true
}
```

Et ensuite lancez `composer requires glpi-project/tools`.

Dans le fichier de configuration de `travis`, ajoutez juste l'appel au CS :

```
script:
  - vendor/bin/robo --no-interaction code:cs
```

Dans le fichier `.gitignore`, ajoutez ce qui suit :

```
dist/
vendor/
.gh_token
*.min.*
```

Quant au fichier de configuration de `Robo.li`, notez que celui qui est fourni par le plugin `empty` est quelque peu spécifique, vous devrez le modifier pour que tout fonctionne. Voir ci-dessous pour plus d'informations.

Pour finir, comme le projet `tools` fournit quelques fonctionnalités, vous pouvez supprimer les scripts utilitaires en doublon (les fichiers tels que `release`, `extract_template.sh`, ...) qui pourraient être présents dans votre plugin.

## 2.3 Fonctionnalités

### 2.3.1 Standards de codage

Les règles GLPI `PHPCodeSniffer` sont fournis en tant que `vendor/glpi-project/coding-standard/GlpiStandard/`.

Pour vérifier les standards, utilisez la tâche `code:cs` de `Robo.li` :

```
$ ./vendor/bin/robo code:cs
```

---

**Note :** La commande ci dessus ignorera `vendor` et sera lancée dans le répertoire courant.

Si vous souhaitez adapter la liste d'ignorance ou les dossiers vérifiés ; vous pouvez simplement surcharger `$csignore` et/ou `$csfiles` dans le `RoboFile.php` du plugin :

```
<?php

class RoboFile extends Glpi\Tools\RoboFile
{
    protected $csignore = ['/vendor/', '/lib/'];
    protected $csfiles  = ['./', 'setup.php.tpl']
    [...]
}
```

## 2.3.2 Vérifications automatiques

Pour des raisons pratiques, un fichier `.travis.yml` est également fourni, il est paramétré pour :

- vérifier les standards de codage,
- être lancé sur un certain nombre de configurations différentes

Vous devrez cependant activer les constructions depuis travis-ci depuis leur site web pour que les tests automatiques soient actifs.

Bien entendu, le fichier `.travis.yml` peut être adapté; vous pouvez lancer des tests unitaires, créer/modifier une base de données, activer les notifications, etc. Référez-vous à la [documentation Travis-CI](#) pour plus de détails.

## 2.3.3 Minifying CSS and JS

**Avertissement :** Désactivé à partir de la 0.1.13, les bibliothèques utilisées par Robo sont anciennes et ne peuvent être remplacées.

Un script d'agrément, qui utilise [Robo.li](#) est fourni. Le fichier `RoboFile.php` est une classe vide qui étends `GlpI\Tools\RoboFile` (fournie par la dépendance `glpi-project/tools`) dans laquelle vous pouvez définir ce que vous voulez.

De cette manière, vous pourrez facilement tenir à jour le fichier commun sans que vos tâches spécifiques ne soient affectées.

Pour installer les bibliothèques requises, vous devrez [installer composer](#) puis lancer :

```
$ composer install -o
```

Il y a 3 cibles disponibles :

- `minify` qui va minimifier tous les fichiers CSS et JS (voir ci-dessous),
- `minify:css` qui va minimifier toutes les feuilles de style CSS dans le dossier `css` de votre plugin, en créant une version `.min.css` version à côté du fichier original,
- `minify:js` qui va minimifier tous les fichiers javascript dans le dossier `js` de votre plugin, en créant une version `.min.js` version à côté du fichier original.

Choisissez simplement une cible, et lancer comme ceci :

```
$ ./vendor/bin/robo minify:css
```

---

**Note :** Souvenez-vous que les fichiers compilés ne doivent pas être commités sur votre dépôt de sources. Leur création fait partie du travail du processus de release.

Pensez également à adapter vos scripts pour qu'ils chargent les versions minimifiées si elles sont disponibles, et l'original dans le cas contraire :)

À partir de GLPI 9.2, vous n'avez plus à vous soucier du chargement des fichiers minimifiés lorsque vous utilisez les hooks `add_css` et `add_javascript` ! Vous devez simplement appeler vos fichiers standards, et GLPI utilisera automatiquement la version minimifiée si elle existe, et si le mode *DEBUG* n'est pas actif.

---

### 2.3.4 Traductions

GLPI et ses plugins utilisent gettext pour l'internationalisation. Plusieurs étapes sont requises pour que cela fonctionne :

1 les chaînes à traduire doivent être extraites depuis les fichiers sources, un fichier POT sera ainsi créé ou mis à jour en conséquence, 2 les fichiers PO doivent être créés ou mis à jour depuis les fichiers POT, 3 les fichiers PO doivent être traduits, 4 les fichiers ``.mo`` doivent être compilés depuis la dernière version des PO.

Dans le dossier `vendor/bin`, vous trouverez un script `extract_template.sh`. Il va extraire les chaînes à traduire depuis le code source (voir le premier point ci-dessus).

Une fois qu'il aura été lancé, le fichier `locale/mygreatplugin.pot` sera créé ou mis à jour.

Pour les secondes et troisièmes étapes, vous aurez à faire un choix. Vous pouvez utiliser l'outil gettext pour mettre à jour vos fichiers PO et les traduire en utilisant un outil dédié, tel que [poedit](#) ; ou vous pouvez utiliser un système de traduction en ligne tel que [Transifex](#) ou [Zanata](#). Le cœur de GLPI ainsi que de nombreux plugins sont traduits via Transifex actuellement.

Une fois que vos fichiers PO auront été mis à jour, vous devrez les compiler en tant que fichiers MO. Vous pouvez lancer cela manuellement, le script de release le fera de toutes façons ; référez-vous à la [section relative à la compilation des fichiers MO](#).

### 2.3.5 Script de release

Un script de release est fourni dans `vendor/bin/plugin-release`. C'est un « simple » script [Python](#) ; vous devez juste avoir Python installé sur votre système (c'est le cas par défaut sur la plupart des distributions linux).

**Avvertissement :** Notez que ce script de release n'est compatible que si vous utilisez le versionnage sémantique !

En utilisant les options par défaut, le script va tenter de retrouver le dernier tag de votre dépôt git, ajouter les dépendances tierces et créer une *Release* sur votre projet github.

```
$ ./vendor/bin/plugin-release
Do you want to build version 1.9.5? [Yes/no] y
Building glpi-order-1.9.5...
Archiving GIT tag 1.9.5
Adding vendor libraries
$ ls dist
glpi-order-1.9.5.tar.bz2  glpi-order-1.9.5.tar.bz2.asc
```

### Pré-requis

Vous aurez besoin d'un interpréteur python installé ainsi que les modules suivants :

- [termcolor](#),
- [gitdb](#),
- [github](#) (pour vérifier les versions existantes dans les brouillons également, et pour créer les releases github), à moins que vous ne spécifiez l'option `--nogithub`

Si vous voulez de l'aide sur le script, essayez de lancer `./vendor/bin/plugin-release -h`.

### Processus

Le processus de release effectuera les tâches suivantes pour vous :

- vérifie que la constante de version soit identique au tag demandé ;
- vérifie que la version dans le fichier XML pour le site web est la même que celle du tag demandé ;
- vérifie si une release existe déjà en local ou en distant (en présumant que votre plugin soit hébergé sur l'organisation github *pluginsGLPI* et qu'elle soit publique) ;
- crée une archive des chemins qui ne sont pas exclus (*.git*, *tools*, *tests*, ...);
- s'il y en a, installe les dépendances ;
- s'il y en a, compile vos fichiers MO ;
- s'il y en a, compile vos feuilles de style CSS et vos fichiers Javascript (en utilisant [Robo.li](#)) ;
- crée une archive de release avec tout cela ; et la met à disposition dans le dossier *dist* ;
- utilise GPG pour signer l'archive ;

**Note :** Le processus de release standard ne travaillera pas sur vos fichiers directement, il effectuera en premier lieu une copie dans le dossier *dist/src* auparavant. La seule exception concerne l'option de *compilation des fichiers MO*.

Dans le but de vérifier si tout est OK avant de créer la vraie release ; créez votre tag et lancez `./vendor/bin/plugin-release -C` **avant de le pousser sur votre dépôt distant**. De cette manière, vous serez en mesure de corriger les éventuels problèmes et re-créez votre tag en local (n'oubliez pas que les tags publiés ne devraient **jamais** être supprimés).

## Compilation des fichiers MO

Le processus de release compilera chaque fichier PO trouvé dans le répertoire *locales*. Vous souhaiterez certainement que les derniers fichiers MO soient présents dans els sources, à des fins de tests. Le script de release fournit l'option `--compile-mo` (ou `-m`) pour cela.

```
$ ./vendor/bin/plugin-release --compile-mo
```

**Avertissement :** La commande ci-dessus travaillera directement sur vos fichiers et non sur une copie, à l'inverse des autres commandes.

## Pré-releases

Par défaut, le script de release ne travaillera que sur des tags existants. Toute pré-release devrait être dûment taguée ; mais il est possible que vous souhaitiez créer une archive sans tag dans certaines circonstances.

Pour indiquer au script de release ce qu'il doit archiver, vous devrez spécifier plusieurs paramètres :

- `--commit` (or `-c`) fournissant le hash du commit,
- `--release` (ou `-r`) fournissant la version (habituellement, la version à venir),
- `--extra` (or `-e`) pour préciser une chaîne complémentaire (telle que *alpha*, *beta*, *rc1*, etc...)

Par exemple pour le plugin *order* :

```
$ ./vendor/bin/plugin-release --commit 632d515d4ac0 --release 1.9.5 --extra alpha1
$ ls dist
glpi-order-1.9.5-alpha1-20161103-632d515d4a.tar.bz2
```

## Signature des releases

La signature des releases avec une clé GPG permettra aux utilisateurs de vérifier l'intégrité du téléchargement avant l'installation. Vous aurez besoin d'une clé GPG diffusée publiquement ; l'option *signature* est active par défaut, vous pouvez la désactiver en utilisant l'option `--nosign` (ou `-S`).

Un fichier contenant la signature portant le même nom que l'archive avec une extension `.asc` sera créé dans le dossier `dist`.

### Release GitHub

Le script de release créera une release GitHub sur votre dépôt, en brouillon, à moins que vous ne spécifiez l'option `--nogithub` (ou `-g`).

---

**Note :** Malheureusement, je n'ai pas été en mesure de téléverser la nouvelle archive sur cette release... Peut-être cela sera-t-il fixé dans le futur.

---

Pour pouvoir bénéficier de cette fonctionnalité, il faudra que le [module python github](#) soit installé ; et il vous faudra un token d'accès. Les tokens d'accès sont valides par utilisateur et donnent accès à l'ensemble de ses dépôts.

Vous devrez vous rendre sur [la page des préférences de votre compte github](#), dans l'onglet « *personal access token* ». Clic sur *generate new token*, fournissez la description souhaitée et assurez-vous d'avoir coché la case *public\_repo* uniquement (il n'est pas nécessaire de cocher quoi que ce soit d'autre, vous pourrez créer plusieurs tokens au besoin).

Le token ne sera affiché qu'une fois ; enregistrez-le dans le fichier `.gh_token` dans le répertoire de votre plugin, et c'est tout !

### Exclure des fichiers

Vous pouvez créer un fichier `.ignore-release` à la racine de votre plugin et y lister les fichiers et dossiers que vous voulez exclure explicitement de l'archive. Écrivez une [expression](#) par ligne

```
.\.png  
screenshots/
```

Certains fichiers seront automatiquement exclus :

- `.git*`,
- `.gh_token`
- `.tx/`
- `tools/`
- `tests/`
- `.atoum.php`
- `.travis.yml`
- `.circleci/`
- `.ignore-release`
- `composer.lock`
- `Robofile.php`





## CHAPITRE 3

---

### Exemple

---

- Lien des sources : <https://github.com/pluginsGLPI/example>
- Téléchargement : *non pertinent*

Un plugin d'exemple... Prévu pour vous montrer les différentes possibilités qui sont offertes aux plugins depuis le cœur de GLPI.

Comme ce plugin ne fait rien en réalité, il n'est pas documenté. Vous souhaitez certainement vous référer à la *présentation des plugins*, au *manuel de développement des plugins* ou encore au *plugin « empty »* comme point de départ.





- Sources link : <https://github.com/pluginsGLPI/treeview>
- Download : <https://github.com/pluginsGLPI/treeview/releases>

### 4.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

### 4.2 Features

This plugin allows you to browse through your inventory via a tree view in a panel located on the left side of the screen.

You can browse the tree of your site to see the equipment.

### 4.3 Install the Plugin

- Uncompress the archive.
- Move the `tag` directory to the `<GLPI_ROOT>/plugins` directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

### 4.4 Usage

The plugin will create required tables in the database automatically. Those tables will be updated along with the plugin.

Plugin configuration is quite simple :

You will access the create form from the *Setup > Plugins > treeview*.

The screenshot shows the GLPI web interface. The top navigation bar includes 'Assets', 'Assistance', 'Management', 'Tools', 'Administration', and 'Setup'. The 'Setup' menu is expanded, showing 'Home' and 'Plugins'. The 'Plugins' sub-menu is active, displaying 'Root entity (tree structure)'. The main content area is titled 'Display' and contains several configuration options for the treeview plugin:

- Target for all the nodes: ...entral console
- Should folders be links: Yes
- Nodes can be highlighted: Yes
- Tree is drawn with lines: Yes
- Tree is drawn with icons: Yes
- Only one node within a parent can be expanded at the same time: No
- Item name: ...ory number / Name
- Location name: ...name / Comment

A 'Post' button is located at the bottom right of the configuration form. At the bottom of the page, a footer indicates 'GLPI 9.4.3 Copyright (C) 2015-2019 Teclib' and 'FusionInventory 9.4.2.3 - Copyright (C) 2010-2019 by FusionInventory Team'.

- Target for all node : on clic on node, redirect to new window
- Nodes can be highlighted : highlight selcted node or not
- Tree is drawn with lines : display or not
- Tree is drawn with icons : Display or not icon for node
- Only one node within a parent can be expanded at the same time : more tthan one parent node can be expand at same time
- Item name : name or inventory number or name + inventory number or inventory number + name
- Location name : short name or long name or long name + comment or long name + comment



- Sources link : <https://github.com/pluginsGLPI/tag>
- Download : <https://github.com/pluginsGLPI/tag/releases>

## 5.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

## 5.2 Features

- Add tags on every GLPI item type (which extends CommonDBTM)
- Add tags on FusionInventory Task (plugin fusioninventory)
- Add tags on Appliance (plugin appliance)
- Manage tag from labels
- Tags are displayed and editable in the object form header (just below title)
- You can search for tags and display them in objects lists
- You can display a report from associated objects in administration

## 5.3 Install the Plugin

- Uncompress the archive.
- Move the tag directory to the <GLPI\_ROOT>/plugins directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

## 5.4 Usage

The plugin will create required tables in the database automatically. Those tables will be updated along with the plugin.

Plugin usage is quite simple :

— Create tag

You will access the create form from the *Setup > Plugins > tag*.

Define name, color, sub-entities visibility and associated itemtype

The screenshot shows the 'New Item - Tag' form in the GLPI interface. The form is titled 'New Item - Tag' and has a 'Root entity' dropdown set to 'No'. The 'Sub-entities' dropdown is also set to 'No'. The form contains the following fields:

- Name: Tag 1
- Description: (empty text area)
- HTML color: A color picker showing red.
- Associated item Types: A button labeled 'Assets - Computers'.

At the bottom right, there are two buttons: 'Add' and 'Cancel'.

— Use tag

On associated itemtype, you can add tag previously created

The screenshot shows the 'test (Root entity)' form in the GLPI interface. The form is titled 'test (Root entity)' and has a 'Root entity' dropdown set to 'No'. The 'Sub-entities' dropdown is also set to 'No'. The form contains the following fields:

- Tags: A list of tags on the left, with 'Tag 1' selected.
- Name: (empty text field)
- Location: (empty text field)
- Type: (empty dropdown)
- Manufacturer: (empty text field)
- Model: (empty text field)
- Serial Number: (empty text field)
- Inventory/Asset Number: (empty text field)
- Network: (empty text field)
- Group: (empty dropdown)
- Domain: (empty text field)
- UUID: (empty text field)
- Update Source: (empty text field)

At the bottom right, there are two buttons: 'Save' and 'Put in Trash Bin'.

— Search items by tag

You can easily find items with tag

The screenshot shows the GLPI interface with the 'Computers' tab selected. A search bar at the top contains 'Tag 1'. Below the search bar, there is a table with columns: Name, Entity, Status, Manufacturer, Serial Number, Type, Model, Operating System - Name, Last Update, and Plugins - Tags. The table contains one row with the name 'test' and the entity 'Root entity'. The 'Last Update' column shows '2020-01-22 08:34'. The 'Plugins - Tags' column shows 'Tag 1'.

— Get all items associated to tag  
Go to tag form, clic on tab « Associated items »

The screenshot shows the GLPI interface with the 'Tag 1 (Root entity)' form. The 'Associated items' tab is selected. The form displays a table with columns: Type, Name, Entity, Serial Number, and Inventory/Asset Number. The table contains one row with the type 'Computer', the name 'test', and the entity 'Root entity'.







- Sources link : <https://github.com/pluginsGLPI/news>
- Download : <https://github.com/pluginsGLPI/news/releases>

## 6.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

## 6.2 Features

This plugin allows to display alerts messages on GLPI homepage and / or login page.

- Create alert with rich text as for notes.
- Choice of begin/end publication date.
- Manage alerts by entity.
- Target : display alerts by profile / group / user.

## 6.3 Install the Plugin

- Uncompress the archive.
- Move the news directory to the <GLPI\_ROOT>/plugins directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

## 6.4 Usage

The plugin will create required tables in the database automatically. Those tables will be updated along with the plugin.

Plugin usage is quite simple :

- configure alerts

You will access the plugin configuration from the *Tools > Alerts*.

## 6.5 Create alerts

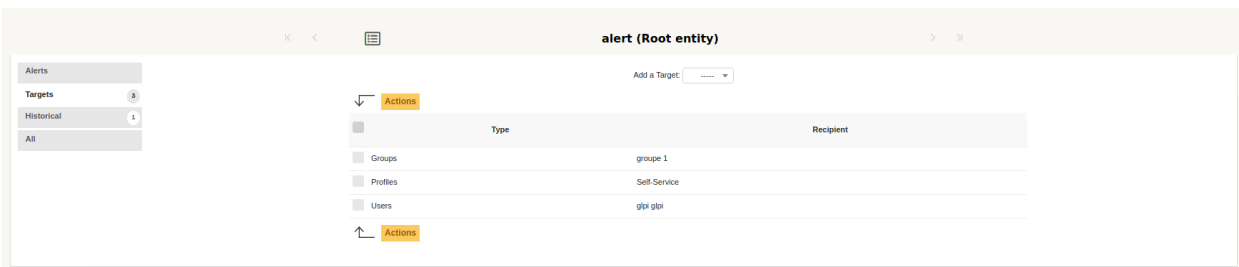
- Name : display as talert title
- Active : is active or not
- Description : content of alert
- Visibility start / end : period of Visibility
- Type : type of alert (General, Information, Warning, Problem)
- Can close alert : User can close alert ?
- Show on login page : display or not alert on login page
- Show on helpdesk page : display or not alert on helpdesk page
- Sub-entity : display or not on sub-entity

The screenshot shows the 'New Item - Alerts' configuration form. At the top, there are tabs for 'New Item - Alerts', 'Root entity', and 'Sub-entities', with a dropdown menu set to 'No'. The form fields include: 'Name' (text input), 'Active' (dropdown set to 'No'), 'Description' (rich text editor with a toolbar), 'Visibility Start Date' (calendar icon), 'Visibility End Date' (calendar icon), 'Type (to add an icon before alert title)' (dropdown), 'Can close alert' (dropdown set to 'Yes'), 'Show on login page' (dropdown set to 'No'), and 'Show on helpdesk page' (dropdown set to 'No'). An 'Add' button is located at the bottom center.

## 6.6 Targets

You can define target by type

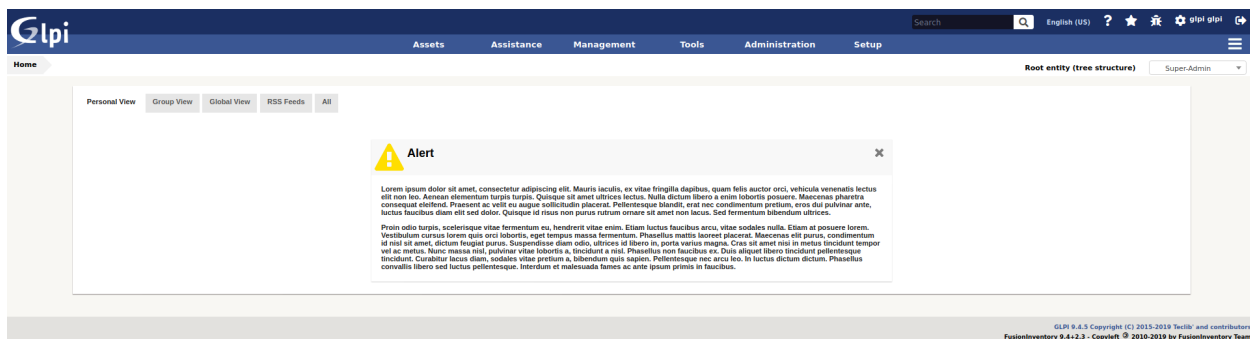
- Group : for all users of group
- Profil : for all users with profile
- User : for specifi user



## 6.7 Display on login page



## 6.8 Display on helpdesk page





- Sources link : <https://github.com/pluginsGLPI/sccm>
- Download : <https://github.com/pluginsGLPI/news/sccm>

## 7.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4
- FusionInventory plugin for GLPI
- SCCM >= 1802
- PHP curl\_init and sqlsrv\_connect
- Microsoft System Center Configuration Manager
- Microsoft Drivers for PHP for Microsoft SQL Server

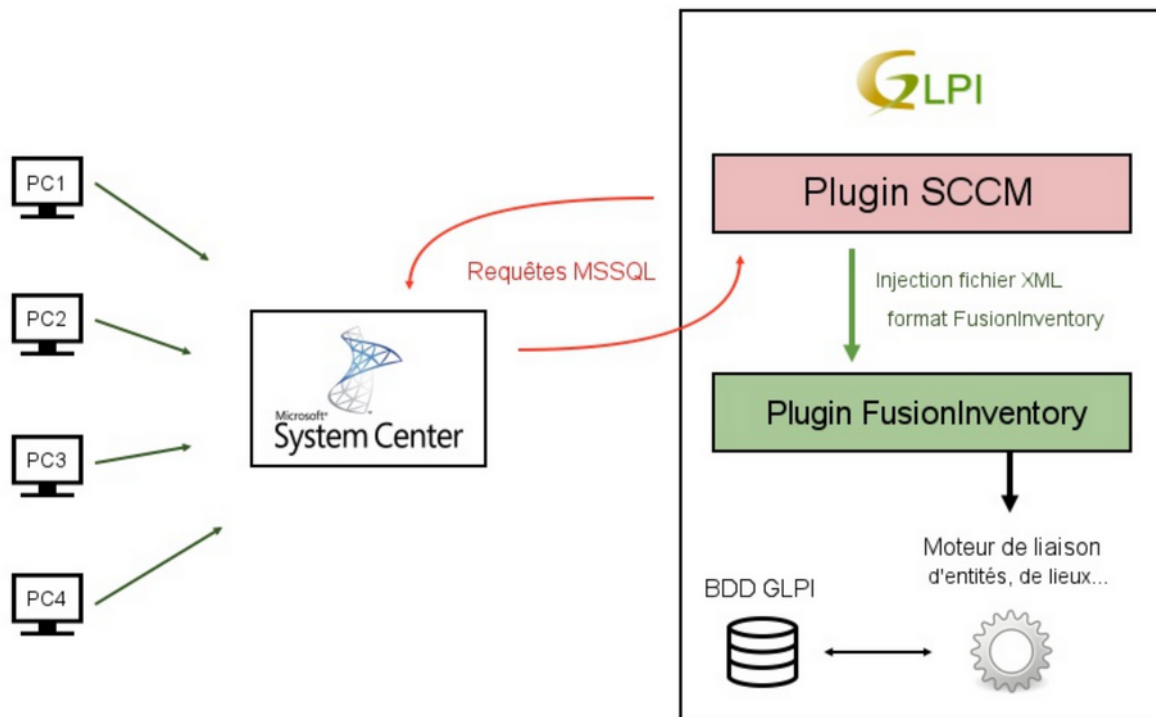
## 7.2 Features

Plugin to synchronize computers from SCCM (version 1802) to GLPI (version 9.3 and 9.4). It uses the « FusionInventory for GLPI » plugin and the power of its internal engine.

## 7.3 Workflow

- This plugin add two automatic actions : « SCCMCollect » et « SCCMPush »
- The automatic action « SCCMCollect » queries the SCCM server with MsSQL queries.
- This same action builds an XML foreach computer (in FusionInventory format).
- The automatic action « SCCMPush » injects XML files into GLPI over HTTP(s) (via cURL and FusionInventory) to display computer in GLPI.

## 7.4 Schematic diagram

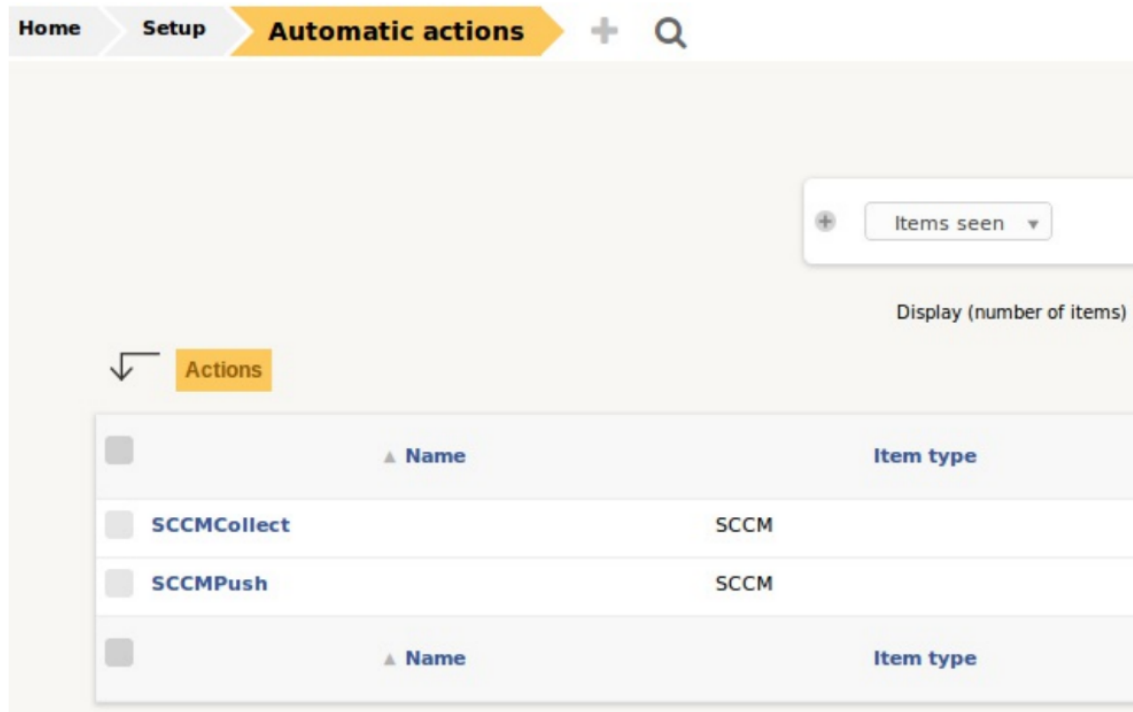


## 7.5 Collected data

Depending on data availability in the SCCM server, foreach computer we collect :

- Computer : name, last connected user, UUID, user login, domain, comments, serial
- Operating system : name, architecture, version, service pack
- BIOS information : tag, model, type, manufacturer, serial, install date, version
- CPU : description (or name), manufacturer, frequency, type (arch), nb core, nb threads
- Softwares : name, version, editor, install date
- Memory (RAM) : capacity, decription, frequency, type, bus, serial
- Graphical card : name, chipset, memory
- Sound card : name, manufacturer
- Network card : name, MAC address, IP address (v4, v6)
- Disk part : name, mount type, total size

## 7.6 Automatic actions



### 7.6.1 SCCMCollect

Action to collect data from SCCM to XML files.

### 7.6.2 SCCMPush

Action to push XML file to FusionInventory plugin







- Sources link : <https://github.com/pluginsGLPI/escalade>
- Download : <https://github.com/pluginsGLPI/escalade/releases>

## 8.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

## 8.2 Features

Escalade plugin aims to simplify the ticket escalation process in GLPI. It also adds a graphical historical for assigned groups.

## 8.3 Install the Plugin

- Uncompress the archive.
- Move the `escalade` directory to the `<GLPI_ROOT>/plugins` directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

## 8.4 Usage

The plugin will create required tables in the database automatically. Those tables will be updated along with the plugin. Plugin usage is quite simple :

— configure plugin

You will access the plugin configuration from the *Setup > Plugins > Escalade*.

Configuration Escalade plugin			
Remove old assign group on new group assign	Yes ▾	show group assign history visually	Yes ▾
Escalation history in tasks	Yes ▾	Remove technician(s) on escalation	Yes ▾
Ticket status after an escalation	Don't change ▾	Assign ticket to initial group on solve ticket	Yes ▾
Assign the technical manager on ticket category change	No ▾	Assign the technical group on ticket category change	No ▾
Clone tickets	Yes ▾	Close cloned tickets at the same time	Yes ▾
Use the technician's group	No ▾		
Display delete button		<b>Requester</b>	<b>Watcher</b>
		<b>User</b>	Yes ▾
		<b>Group</b>	Yes ▾
		<b>Supplier</b>	Yes ▾
Enable filtering on the groups assignment	No ▾		
<b>Save</b>			

## 8.5 Configuration

— Remove old assign group on new group assign

When this option is enabled, plugin remove old group when new group is assign to ticket

— Show group assign history visually

When this option is enabled, plugin display display group assign history

Actor	Requester +	Watcher + ⓘ	Assigned To + ⓘ
	normal normal ⓘ ⓘ		<div> groupe 1 - ⓘ ⓘ </div> <div> groupe 2 </div> <div> groupe 3 </div> <div> groupe 2 </div> <div> groupe 1 </div>

— Escalation history in tasks

When this option is enabled, the plugin adds a private task containing the escalation information fore each escalation

Ticket
Processing Ticket 5
Statistics
Approvals
Knowledge Base
Items
Costs

test (Root entity)

Add :
Followup
Task
Document
Solution

Actions Historical :

2020-01-21 11:25

Gipi Gipi ⓘ

escalated to the group groupe 1

— Remove technician(s) on escalation

When this option is enabled, the plugin remove assign technician when escalation is performed

— Ticket status after an escalation

When this option is enabled, the plugin change ticket status when escalation is performed

- Assign ticket to initial group on solve ticket

When this option is enabled, the plugin assign first group assigned to ticket

- Assign the technical manager on ticket category change

When this option is enabled, the plugin assign first group assigned to ticket

- Assign the technical group on ticket category change

When this option is enabled, the plugin assign the technical group of category selected from ticket

- Clone tickets

When this option is enabled, the plugin gives the possibility to clone a ticket and to link it the initial ticket

- Close cloned tickets at the same time

When this option is enabled, at the closing of a ticket, cloned and linked tickets are also closed

- Use the technician's group

When this option is enabled, the plugin assign first or latest group of technician assign to ticket

- Display delete button

Display or not delete button for group, user or supplier switch type (requester, watcher, assigned to)

- Enable filtering on the groups assignment

When this option is activated, the plugin filters the groups displayed in the dropdown list

- Example

- “glpi” user is affected to group “groupe 1”

- “groupe 1” is configured to allow ticket escalation only to “group 2”

- “glpi” user can only reassign ticket to “groupe 2” because it’s assign to “groupe 1”

Urgence: Moyenne  
Impact: Moyen  
Priorité: Moyenne  
Validation: Non soumis à validation  
Lieu: TEST 1  
Acteur: Demandeur +, Observateur +  
groupe 1 - i

— For other users not assign to “groupe 1”, they can’t reassign group to ticket

Acteur: Demandeur +, Observateur +, Attribué à + i  
Titre: test  
Groupe: groupe 1 - i

— You can « bypass » this behavior per user

Utilisateur: groupe 1  
Bypass filtering on the groups assignment: Non  
AJAX DEBUG: Display only tabs for debug

## 8.6 Ticket’s search engine

Plugin add three new search option

...characteristics - Status is Not Solved  
rule global  
Problems  
Number of Problems  
Documents  
Number of documents  
Plugins  
Group concerned by the escalation  
Number of escalations  
Number of escalations between two groups  
Entity: Root entity, Status: Processing (assigned), Last Update: 2020-01-21 14:07, Opening Date: 2020-01-21 10:16, Prior: Medium

— Group concerning by escalation

Ex : get all tickets actually escalated to group « groupe 1 »

— Number of escalation


Ex : get all tickets with more than ten escalation

— Number of escalation between two groups


Ex : get all tickets with more than 2 escalations between two groups

## 8.7 Central page

New dashlet to display ticket solved to close and assign to you group

Tickets to close (climbed) <span>1</span>			
	Requester	Associated Element	Description
ID: 31	normal normal 	General	test (0 - 10)

New dashlet to display not solved ticket where your groups was not actually assign but was in the past

Tickets to follow (climbed) <span>1</span>			
	Requester	Associated Element	Description
ID: 31	normal normal 	General	test (0 - 10)





- Sources link : <https://github.com/pluginsGLPI/credit>
- Download : <https://github.com/pluginsGLPI/credit/releases>

### 9.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

### 9.2 Features

This plugin allows you to declare and follow (by entity) the consumptions of different credit vouchers directly from the Ticket form.

To do it, you must follow these steps :

- Creation of different types of credit vouchers (as GLPI dropdown);
- Creation, in Entity tab, of the credit vouchers available for these entities;
- Declaration / Modification / Deletion of a consumption in the Ticket form and tab;
- Consultation of the vouchers sold, the remaining credits, credits consumptions from the Ticket form or from Entity tab.

### 9.3 Configuration

The plugin will create required tables in the database automatically. Those tables will be updated along with the plugin.

Plugin usage is quite simple :

You will access the credit configuration from the *Administration > Entities*.

The screenshot shows the 'Ajouter un crédit' form in the GLPI Administration interface. The form is titled 'Root entity ( + Sub-entities)' and includes the following fields:

- Name:
- Type:
- Start Date:
- End Date:
- Active: ☐
- Quantity:
- Allow overconsumption: ☐

Below the form, it says 'Aucun crédit déclaré' (No credit declared). The sidebar on the left lists various entities like Entities, Address, Notifications, etc. The bottom of the page shows a copyright notice for GLPI 9.4.5.

- Name : add name to credit
- Type : Define a type
- Start / End date : Define duration of credit
- Active : Display credit on ticket form or not
- Quantity : quantity spend
- Allow overconsumption : allow or not over consumption (if yes, check is bypass)

## 9.4 Credit consumption

When you solved a ticket, solution form have field to to consume credit

The screenshot shows the 'New Item - Solution' form in the GLPI Tickets interface. The form includes the following fields:

- Solution Template:
- Solution Type:
- Save & Add to the Knowledge Base: ☐
- Description:
- Crédits section:
  - Consommation de crédit: ☐
  - Nom du crédit:
  - Quantité consommée:

The sidebar on the left lists various tickets like Processing Ticket, Statistics, Approvals, etc. The bottom of the page shows a timeline filter.

## 9.5 Credit report

On the ticket form, click on the *Credit* tab to view a report on credit consumption



GLPI

Search

English (US)

?

★

⚙️

glpi glpi

⌵

Assets

Assistance

Management

Tools

Administration

Setup

Home

Assistance

Tickets

+

🔍

⌵

🔊

Root entity (tree structure) Super-Admin

Ticket

Processing Ticket 1

Statistics

Approvals

Knowledge Base

Items

Costs

Projects

Project Tasks

Problems

Changes

Historical 6

Credit 1

All

test (Root entity)

Credits déjà consommés pour ce ticket

Nom du crédit	Type de crédit	Date de consommation	Utilisateur déclarant	Quantité consommée
Credit		2020-01-23	glpi glpi	4

Credits actifs pour l'entité du ticket

Name	Type	Active	Start Date	End date	Quantité vendue	Quantité consommée	Quantité restante	Allow overconsumption
Credit		Yes			10	4	6	No

GLPI 9.4.3 Copyright (C) 2013-2019 Teclib' and contributors

FusionInventory 9.4.2.3 - Copyright © 2010-2019 by FusionInventory Team





# CHAPITRE 10

---

## Generic Object

---

- Lien des sources : <https://github.com/pluginsGLPI/genericobject>
- Téléchargement : <https://github.com/pluginsGLPI/genericobject/releases>

This user manual applies to version 2.5 of the GLPI Generic Object Plugin.

### 10.1 Pré-requis

Please refer to *plugin page* <<https://plugins.glpi-project.org/#/plugin/genericobject>>.

### 10.2 Fonctionnalités

Ce plugin vous permet d'ajouter de nouveaux type d'objets qui seront intégrés dans GLPi

Il supporte les fonctionnalités suivantes de GLPi :

- gestion des Entités et sous-Entités ;
- recherche ;
- modèles ;
- historique ;
- intégration dans l'assistance ;
- intégration dans le plugin d'injection de fichiers CSV ;
- intégration dans le plugin de désinstallation d'objets ;
- intégration dans le plugin de gestion des commandes.

### 10.3 Exemple d'utilisation

**Objectif :** gérer votre flotte de véhicules de la même manière que le reste de votre parc.

- Créer un nouvel objet d'inventaire de type *véhicule*.

- Ajouter les champs adéquats pour un *véhicule*, comme : *nom*, *numéro de série*, *numéro d'inventaire*, *type*, *modèle*, *couleur*, *état*, etc.
- Décrire le comportement d'un *véhicule* : visible dans les sous entités, conserver l'historique, etc.
- Ajuster les droits sur *véhicules*.
- Activer l'objet *véhicules*.
- Gérer votre collection de *véhicules* dans GLPI.

## 10.4 Installer le plugin

- Décompresser l'archive.
- Déplacer le répertoire `genericobject` dans le répertoire `<GLPI_ROOT>/plugins`.
- Se rendre sur la page *Configuration > Plugins*
- Installer et activer le plugin

## 10.5 Utilisation

### 10.5.1 Créer un nouveau type d'objet

Ceci constitue la première étape.

- Cliquer sur le bouton + dans le formulaire de configuration du plugin.
- Créer un nouvel objet de type inventaire :
  - *nom\** : requis, en minuscules, et doit être composé de lettres uniquement ;
  - *libellé* : par défaut, identique au nom.
- Valider.
- Activer le nouveau type d'objet pour l'utiliser.

**Exemple** : créer un nouveau type d'objet d'inventaire *véhicule*.

### 10.5.2 Modifier les libellés

Pour chaque type, un fichier de langue est disponible dans `<GLPI_ROOT>/files/_plugins/genericobject/locales/itemtype/`

Le plugin créé :

- un fichier de langue pour la langue courante
- un fichier de langue pour la langue par défaut de GLPI

---

**Note** : Si la langue courante et la langue par défaut sont identiques, un seul fichier sera créé.

---

Pour modifier le libellé du type d'objet, pour la langue anglaise, éditer le fichier :

```
<?php
// <GLPI_ROOT>/files/_plugins/genericobject/locales/<itemtype>/<itemtype>.en_GB.php
$LANG['genericobject']['<itemtype>'][1] = "<type's label>";
```

### 10.5.3 Configuration du comportement

**Exemple** : décrire le comportement d'un *véhicule* : visible dans les sous entités, conserver l'historique, etc.

Le nouveau type sera géré de la même manière que les types usuels de GLPI (ordinateurs, moniteurs, matériels réseau, etc.)

---

**Note :** Tous les objets sont au moins assignés à une *entité*

---

L'onglet comportement vous permet de définir :

- *sous-entités* : autorise le type à être récursif ;
- *Assistance* : autorise l'objet à être associé à un ticket ;
- *Corbeille* : utiliser la fonctionnalité corbeille de GLPI ;
- *Notes* : utiliser la fonctionnalité notes de GLPI ;
- *Historique* : autorise l'historique pour ce type ;
- *Modèles* : autorise la gestion de modèles ;
- *Documents* : permet à des documents d'être attachés à ce type ;
- *Prêts* : autorise le prêt d'objet ;
- *Contrats* : lie un objet à un ou plusieurs contrats ;
- *Connexions réseau* : permet l'utilisation et la gestion de ports réseau pour ce type ;
- *Plugin d'injection de fichiers CSV* : permet à ce type d'être disponible pour utilisation dans le plugin ;
- *Plugin de désinstallation d'objets* : autorise la désinstallation de ce type ;
- *Plugin de gestion des commandes* : autorise le lien de ce type à une commande ;

### 10.5.4 Ajouter des champs

**Exemple :** ajouter les champs adéquats pour un *véhicule*, comme : *nom, numéro de série, numéro d'inventaire, type, modèle, couleur, état*, etc.

Aller à l'onglet *Champs*.

Le plugin est fourni avec plusieurs champs prêts à l'utilisation :

- Nom
- Type
- Modèle
- Numéro de série
- Numéro d'inventaire
- Utilisateur
- Groupe
- Statut
- Commentaires
- Notes
- Emplacement
- Autre
- Constructeur
- URL
- Date de création
- Date d'expiration
- Catégorie
- Visible dans l'assistance
- Technicien en charge du matériel
- Domaine
- Contact
- Numéro de contact

---

**Note :** L'utilisation de certains comportement entraînera l'ajout automatique de champs dans l'objet :

---

- connexion réseau => emplacement
- prêts => emplacement
- assistance => visible dans l'assistance
- notes => notepad

### Intégration assistance

Pour utiliser un objet dans l'assistance, utiliser la configuration suivante :

- Dans l'onglet *Comportement* : *utiliser l'assistance* doit être défini à **Oui**.
- Si le champ *Utilisateur* est défini, l'objet sera visible dans la liste *Mes objets* (comme un objet dont l'utilisateur est propriétaire).
- Si le champ *Groupe* est défini, l'objet sera visible dans la liste *Mes objets* également (comme un objet dont le propriétaire est un groupe auquel l'utilisateur appartient).
- Si le champ *Assistance visible* est défini et que sa valeur est à **Non** dans l'objet, ce objet ne sera pas visible du tout dans l'assistance.

### 10.5.5 Ajouter de nouveaux champs

---

**Note :** Les nouveaux champs seront disponibles pour tous les types d'objets.

---

- Créer un nouveau fichier nommé `<GLPI_ROOT>/files/_plugins/genericobject/fields/<type>.constant.php`

Par exemple, pour un type *véhicule* le fichier de constantes sera `<GLPI_ROOT>/files/_plugins/genericobject/fields/car.constant.php`.

Notez que la première ligne du fichier doit être la suivante, sans quoi les nouveaux champs n'apparaîtront pas dans la liste :

```
<?php
global $GO_FIELDS, $LANG;
```

- Ajouter les nouvelles définitions de champs.

### 10.5.6 Ajouter un champ liste déroulante simple

```
<?php
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['name']      = $LANG[
    ↪ 'genericobject']["<type's name>"][2];
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['field']    = 'color';
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['input_type'] = 'dropdown';
```

---

**Note :** La variable de langue doit être définie dans le fichier de langues (voir les explications ci-dessus).

---

- Ajouter un champ liste déroulante qui est assigné à une entité :

```
<?php
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['name']      = $LANG[
    ↪ 'genericobject']["<type's name>"][2];
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['field']    = 'color';
```

(suite sur la page suivante)

(suite de la page précédente)

```

$GO_FIELDS['plugin_genericobject_mytypecolors_id']['input_type'] = 'dropdown';
//Does the dropdown take care of entities ? (true/false)
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['entities_id'] = true;
//Can values be recursive ? (true/false, only taking in account if entities_id is set_
↳to true)
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['is_recursive'] = true;

```

### 10.5.7 Ajouter un champ liste déroulante hiérarchique

```

<?php
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['name'] = $LANG[
↳'genericobject']["<type's name>"][2];
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['field'] = 'color';
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['input_type'] = 'dropdown';
//Is it a tree-dropdown, or a simple one ? (true/false)
$GO_FIELDS['plugin_genericobject_mytypecolors_id']['is_tree'] = true;

```

**Note :** Vous pouvez utiliser en même temps les paramètres suivants : *entities\_id*, *is\_recursive*, *is\_tree*.

### 10.5.8 Ajout d'un champ liste déroulante basé sur les objets du coeur de GLPI (utilisateur, emplacement, ...)

```

<?php
$GO_FIELDS['users_id_passengers_id']['name'] = 'Passenger';
$GO_FIELDS['users_id_passengers_id']['input_type'] = 'dropdown';

```

**Note :** Les noms entre crochets ([]) **DOIVENT** commencer par *users\_id* pour être reconnus comme étant des champs basés sur la liste des utilisateurs de GLPI.

Voir le fichier <GLPI\_ROOT>/files/\_plugins/genericobject/fields/field.constant.php pour une liste complète des champs disponibles.

### 10.5.9 Ajouter une liste déroulante globale

Une liste déroulante globale peut être utilisée dans tous les types d'objets. Un bon exemple serait :

```

<?php
$GO_FIELDS['categories_id']['name'] = $LANG['common'][36];
$GO_FIELDS['categories_id']['input_type'] = 'dropdown';
$GO_FIELDS['categories_id']['dropdown_type'] = 'global';

```

Une table pour les catégorie spécifique sera créée pour chaque type d'objet. Le nom de la table et le nom du champ seront concaténés de cette manière :

- table : `glpi_plugin_genericobject_<itemtypename>_category`
- nom du champ : `plugin_genericobject_<itemtype>categories_id`

### 10.5.10 Ajouter un champ entier

```
<?php
$GO_FIELDS['testinteger']['name']      = 'testinteger';
$GO_FIELDS['testinteger']['input_type'] = 'integer';
$GO_FIELDS['testinteger']['min']       = 10; //not mandatory, by default 0
$GO_FIELDS['testinteger']['max']       = 40; //not mandatory, by default 100
$GO_FIELDS['testinteger']['step']      = 3; //not mandatory, by default 1
```

### 10.5.11 Ajouter un champ texte

```
<?php
$GO_FIELDS['mytextfield']['name']      = 'My text field';
$GO_FIELDS['mytextfield']['input_type'] = 'text';
```

Modifié dans la version 2.1.2 : En ajoutant l'argument suivant, vous indiquerez au plugin que ce champ peut être généré automatiquement en utilisant un modèle :

```
<?php
$GO_FIELDS['mytextfield']['autoname'] = true;
```

### 10.5.12 Ajouter un champ Oui/Non

```
<?php
$GO_FIELDS['mybooleanfield']['name']    = 'My boolean field';
$GO_FIELDS['mybooleanfield']['input_type'] = 'bool';
```

### 10.5.13 Ajouter un champ date

```
<?php
$GO_FIELDS['creationdate']['name']      = $LANG['genericobject']['fields'][30];
$GO_FIELDS['creationdate']['input_type'] = 'date';
```

### 10.5.14 Ajouter un champ date et heure

```
<?php
$GO_FIELDS['creationdate']['name']      = $LANG['genericobject']['fields'][30];
$GO_FIELDS['creationdate']['input_type'] = 'datetime';
```

---

**Note :** Si vous ne souhaitez pas que le champ puisse être modifié en utilisant une action de masse, ajouter la ligne suivante à sa définition :

```
<?php
$GO_FIELDS['myfield']['massiveaction'] = false;
```

---



## 10.6 Ajouter des champs globaux

Pour rendre vos champs accessibles depuis tous les types d'objets :

- Créer un fichier nommé `<GLPI_ROOT>/files/_plugins/genericobject/fields/field.constant.php`
- Ajouter les définitions dans ce fichier.

## 10.7 Configurer les droits

Vous pouvez définir des droits d'accès pour chaque type d'objet, pour chaque profil. Les options disponibles sont :

- *droits sur le type : pas d'accès, lecture, écriture.*
- *droits pour associer ce type d'objet aux tickets : oui, non.*

Pour associer les droits, vous pouvez au choix :

- Utiliser l'onglet *Droits* dans le formulaire du *type d'objet*.
- Aller dans *Administration > Profils* pour administrer les droits de chaque profil.

## 10.8 Utiliser le nouveau champ

Activer le nouveau type, il est désormais prêt à être utilisé.

Le nouveau type est disponible pour les utilisateurs depuis le menu *Plugins > Gestion d'objets*.

## 10.9 Exemple d'utilisation de Generic Object comme GMAO

### 10.9.1 But de cette documentation

Présenter une utilisation complète de Generic Object comme GMAO (Gestion de Maintenance Assistée par Ordinateur) dans un environnement biomédical.

À la fin de cette documentation, vous aurez :

- une entité *Biomed* dédiée (sous l'Entité racine)
- contenant des objets *Biomedical* (dans le menu *Parc*)
- avec des champs pré-définis et personnalisés
- gérée par des utilisateurs avec le profil *Admin\_biomed*

### 10.9.2 Étapes

Les étapes suivantes partent du principe que vous avez une habilitation Super-Admin :

- Installer Generic Object sur GLPI (validé avec genericobject >= 0.85-1.0 et GLPI >= 0.90)
- Configuration de Generic Object
- Configuration de GLPI
- Démarrer l'utilisation de Generic Object et GLPI

### 10.9.3 Installer Generic Object sur GLPI

Voir la section *Installer le plugin*.

## 10.9.4 Configuration de Generic Object

### Créer votre type d'objet

Voir la section *Créer un nouveau type d'objet* et utiliser *biomedical* comme identifiant interne. Le libellé sera défini automatiquement à *Biomedical* (avec un *B* majuscule).

Après une connexion/déconnexion, vous verrez le menu *Biomedical* dans le Parc.

### Définir les nouveaux champs du type Biomedical

Ces champs seront utilisables uniquement par les objets du type Biomedical

- Créer un nouveau fichier appelé : <GLPI\_ROOT>/files/\_plugins/genericobject/fields/biomedical.constant.php
- Ajouter le contenu suivant :

```
<?php
global $GO_FIELDS, $LANG;

// CODE CNEH
$GO_FIELDS['plugin_genericobject_cnehcodes_id']['name']      = $LANG['genericobject
↳ ']['PluginGenericobjectBiomedical'][1];
$GO_FIELDS['plugin_genericobject_cnehcodes_id']['field']     = 'cnehcode';
$GO_FIELDS['plugin_genericobject_cnehcodes_id']['input_type'] = 'dropdown';

// REFORME (yes or no)
$GO_FIELDS['reformed']['name']      = $LANG['genericobject']['
↳ 'PluginGenericobjectBiomedical'][2];
$GO_FIELDS['reformed']['input_type'] = 'bool';

// CLASSE CE (3 choix possibles 1,2a ou 2b)
$GO_FIELDS['plugin_genericobject_classeces_id']['name']      = $LANG['genericobject
↳ ']['PluginGenericobjectBiomedical'][3];
$GO_FIELDS['plugin_genericobject_classeces_id']['field']     = 'classce';
$GO_FIELDS['plugin_genericobject_classeces_id']['input_type'] = 'dropdown';

// UF (Unité Fonctionnelle)
$GO_FIELDS['plugin_genericobject_ufs_id']['name']      = $LANG['genericobject']['
↳ 'PluginGenericobjectBiomedical'][4];
$GO_FIELDS['plugin_genericobject_ufs_id']['field']     = 'uf';
$GO_FIELDS['plugin_genericobject_ufs_id']['input_type'] = 'dropdown';

// PRESTATAIRE BIOMED
$GO_FIELDS['plugin_genericobject_prestataires_id']['name']      = $LANG[
↳ 'genericobject']['PluginGenericobjectBiomedical'][5];
$GO_FIELDS['plugin_genericobject_prestataires_id']['field']     = 'prestataire_
↳ biomedical';
$GO_FIELDS['plugin_genericobject_prestataires_id']['input_type'] = 'dropdown';

// TYPE D'EQUIPEMENT BIOMED
$GO_FIELDS['plugin_genericobject_typedequipementbiomeds_id']['name']      = $LANG[
↳ 'genericobject']['PluginGenericobjectBiomedical'][6];
$GO_FIELDS['plugin_genericobject_typedequipementbiomeds_id']['field']     = "type d
↳ équipement biomedical";
$GO_FIELDS['plugin_genericobject_typedequipementbiomeds_id']['input_type'] = 'dropdown
↳ ';
```

(suite sur la page suivante)

(suite de la page précédente)

```
// Criticite
$GO_FIELDS['plugin_genericobject_criticités_id']['name']      = $LANG['genericobject
↪'] ['PluginGenericobjectBiomedical'] [7];
$GO_FIELDS['plugin_genericobject_criticités_id']['field']     = 'criticite';
$GO_FIELDS['plugin_genericobject_criticités_id']['input_type'] = 'dropdown';

// Numéro marquage CE
$GO_FIELDS['plugin_genericobject_marquageces_id']['name']     = $LANG['genericobject
↪'] ['PluginGenericobjectBiomedical'] [8];
$GO_FIELDS['plugin_genericobject_marquageces_id']['field']     = 'marquagece';
$GO_FIELDS['plugin_genericobject_marquageces_id']['input_type'] = 'dropdown';

// Classe électrique
$GO_FIELDS['plugin_genericobject_classeelec_id']['name']      = $LANG['genericobject
↪'] ['PluginGenericobjectBiomedical'] [9];
$GO_FIELDS['plugin_genericobject_classeelec_id']['field']      = 'classeelec';
$GO_FIELDS['plugin_genericobject_classeelec_id']['input_type'] = 'dropdown';
?>
```

**Avertissement :** Le `s_id` de fin est obligatoire dans `[plugin_genericobject_field*s_id*]`. Le framework GLPI impose que les champs de clés étrangères se terminent par `s_id`. En base de données, `glpi_plugin_genericobject_fields` est le nom de la table et `id`, sa clé étrangère. Voir la [documentation développeur de GLPI](#).

## Définir les libellés des champs

Voir la section *Modifier les libellés*.

- Éditer vos fichiers de locales, par exemple : `<GLPI_ROOT>/files/_plugins/genericobject/locales/biomedical/biomedical.fr_FR.php`
- Ajouter le contenu suivant à la fin du fichier :

```
<?php
// Fields
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [1]="Code CNEH";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [2]="Réformé";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [3]="Classe CE";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [4]="UF";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [5]="Prestataire Biomed";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [6]="Type d'équipement biomed";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [7]="Criticité";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [8]="Marquage CE";
$LANG['genericobject'] ['PluginGenericobjectBiomedical'] [9]="Classe électrique";
```

## Définir les comportements

Dans le menu *Plugins > Gestion d'objets*, sur l'onglet *Principal*, sélectionner :

- *Élément à la corbeille*
- *Historique*
- *Informations financières et administratives*
- *Documents*
- *Recherche globale*

- *Assistance*
- *Modèles*
- *Contrats*
- *Recherche globale*

Cette action ajoutera des champs pré-définis à votre type d'objet

### Ajouter des champs à votre type d'objet

Dans le menu *Plugins > Gestion d'objets*, sur l'onglet *Champs*, vous pouvez maintenant ajouter des champs au type d'objet Biomedical :

- champs pré-définis (champs natifs à GLPI)
- nouveaux champs (définis dans la section *Définir les nouveaux champs du type Biomedical*)

## 10.9.5 Configuration de GLPI

### Définir le profil *Admin\_biomed*

1. Cloner le profil *Admin*
2. Définissez les droits suivants dans le profil *Admin\_biomed* :
  - *Administration > Profils > Admin\_biomed > Onglet Parc > Désélectionner tout*
  - *Administration > Profils > Admin\_biomed > Onglet Assistance > Association > Matériels associables à un ticket > Biomedical*
  - *Administration > Profils > Admin\_biomed > Onglet Gestion > Sélectionner tout*
  - *Administration > Profils > Admin\_biomed > Onglet Gestion d'objets > Biomedical > Sélectionner tout*

---

**Note :** Avec ces paramètres, les utilisateurs avec le profil *Admin\_biomed* voient uniquement *Biomedical* dans le menu Parc

---

### Définir l'entité *Biomed* et les règles d'habilitations

1. Créer l'entité *Biomed* sous l'Entité *Racine* dans *Administration > Entités*
2. Configurer les règles d'habilitations pour assigner le profil *Admin\_biomed* aux utilisateurs de l'entité *Biomed*

## 10.9.6 Démarrer l'utilisation de Generic Object et GLPI

En tant qu'utilisateur *Admin\_biomed*, vous pouvez créer votre premier objet dans *Parc > Biomedical*.

Pour gagner du temps, définissez les valeurs des nouveaux champs dans *Configuration > Intitulés > Gestion d'objets*.



- Sources link : <https://github.com/pluginsGLPI/order>
- Download : <https://github.com/pluginsGLPI/genericobject/order>

This user manual applies to version 2.5 of the GLPI Generic Object Plugin.

## 11.1 Requirements

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

## 11.2 Features

This plugin supports the following aspects of order management :

- Definition and management of a catalogue of product references
- Order management (with approval workflow)
- Budgets management
- Management of invoices associated with orders

## 11.3 Install the Plugin

- Uncompress the archive.
- Move the `escalade` directory to the `<GLPI_ROOT>/plugins` directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

## 11.4 Configuration

You will access the plugin configuration from the *Setup > Plugins > Orders managment*.

When the plugin is installed and enabled, the following options are available :

Plugin configuration	
Default VAT	No VAT ▼ ⓘ
Use validation process	No ▼
Order generation in ODT	No ▼
Activate suppliers quality satisfaction	No ▼
Display order's suppliers informations	No ▼
Color to be displayed when order due date is overtaken	<span style="background-color: red; border: 1px solid black; display: inline-block; width: 20px; height: 10px;"></span>
Copy order documents when a new item is created	No ▼
Default heading when adding a document to an order	----- ▼ ⓘ
Author group (Default values)	----- ▼ ⓘ
Recipient group (Default values)	----- ▼ ⓘ
Recipient (Default values)	----- ▼ ⓘ
Do not display expired budgets	No ▼
Transmit budget change to linked assets	No ▼
Display account section on order form	No ▼
Set account section as mandatory on order form	No ▼
Use free references	No ▼
Rename documents added in order	No ▼

- Default VAT : indicates the amount of VAT that will be proposed by default in the plugin
- Use the validation process : indicates whether the full validation circuit should be used, or whether anyone can validate applications.
- Purchase order generation : indicates if the tab allowing the generation of a purchase order in ODT format is displayed or not.
- Enable supplier quality : indicates whether the tab for judging supplier satisfaction is displayed once the order is delivered.
- Display purchase order vendor information : Shows whether or not the tab for storing the reference and vendor purchase order number is displayed.
- Colour when the estimated delivery date is exceeded : colour of display in the search engine of an undelivered order for which the estimated delivery date is exceeded
- Copy order documents when a new item is created : Copy order documents to generated item
- Default heading when adding a document to an order : define default heading
- Author group (Default values) : Define default author
- Recipient group (Default values) : Define default Recipient group
- Recipient (Default values) : Define default Recipient
- Do not display expired budgets : yes or no
- Transmit budget change to linked assets : yes or no
- Display account section on order form : yes or no
- Set account section as mandatory on order form : yes or no
- Use free references : yes or no
- Rename documents added in order :yes or no

### 11.4.1 Automatic actions when delivery Item

Automatic actions when delivery	
Item	
Display analytic nature on item form	No ▾
Set analytic nature as mandatory on item form	No ▾
Enable automatic generation	No ▾
Default state	----- ▾ ⓘ
Add order location to item	No ▾
Add billing details to item	No ▾

Generation does not apply to contracts, cartridges and consumables. In order to activate this feature, change the option Activate automatic generation to yes and validate. During the generation, the material will be created with some randomly generated information :

- Default name
- Default Serial Number
- Default inventory number
- If a template is associated with the part number, and in this template the name and serial number are generated, then the settings defined above for these 2 fields do not apply.

The default status will be set when the material is created. The positioning of the status during the creation of the material can lead to a date filling of the financial information. See the Entity Inventory tab.

### 11.4.2 Order lifecycle

You can define your own status for order lifecycle

Order lifecycle	
State before validation	----- ▾ ⓘ
Waiting for validation state	----- ▾ ⓘ
Validated order state	----- ▾ ⓘ
Order being delivered state	----- ▾ ⓘ
Order delivered state	----- ▾ ⓘ
Order paid state	----- ▾ ⓘ
Canceled order state	----- ▾ ⓘ

## 11.5 Creation of a reference catalogue

You will access the reference catalogue from the *Management > Orders > Products references*.

The screenshot shows the 'Product reference' form in the GLPI interface. The form is titled 'Galaxy S10 (Root entity)'. It includes a sidebar with 'Product reference', 'Supplier Detail', 'Documents', 'Historical', and 'All'. The main form has the following fields: Name (Galaxy S10), Active (Yes), Manufacturer (Samsung Electronics Co Ltd), Item type (Phones), Model (S10), Last update (2020-01-27 10:19), Comments, Manufacturer reference, Type (SmartPhone), and Template name. A 'Save' button is at the bottom right.

The creation of product references is the prerequisite for order management. Any product to be ordered must be referenced in the plugin catalog. When creating a product reference, the following fields must be filled in :

- Sub-entities : indicates if the reference will be visible in the current entity only or in the sub-entities as well.
- Name : name of the reference, as it will be displayed when managing an order (required)
- Manufacturer : the manufacturer of the product (not the supplier)
- Type of equipment : indicates what type of equipment we are dealing with
- Type : the type of the product
- Model : the model of the product
- Template : the GLPI template of the product if one already exists.
- comments : additional information on the reference
- Once the reference is created, it is then necessary to indicate from which supplier(s) the product is available, as well as the price associated with it.

It is also possible to :

- Add related documents
- View history

## 11.6 Creating an order

You will access orders from the *Management > Orders > Orders*.

The screenshot shows the 'Phone Device (Root entity)' order form in the GLPI interface. The form is titled 'Phone Device (Root entity)'. It includes a sidebar with 'Order', 'Order items', 'Validation', 'Bill', 'Tickets', 'Documents', 'Notes', 'Historical', and 'All'. The main form has the following fields: Order name (Phone Device), Order number (DFG4564DUJO), Order status, Delivery location, Supplier (Samsung), Contact, Associate to a ticket (No), Date of order (2020-01-27), Type, Budget (2019-2020), Payment conditions, Postage (0.00), VAT Postage (20), Estimated due date (2020-01-27), and Delivery date. A 'Save' button is at the bottom right.

Once the reference catalogue has been created, it is now possible to add material orders. The fields to be filled in are :

- Sub-entities : indicates if the order is visible in the sub-entities
- Order name



- Order number (required)
- Date of order (required)
- Budget : The budget to which the purchase order is posted.
- Vendor purchase order number : Indicates the internal number for the purchase order at the vendor.
- Payment terms : free field (can contain for example 30 days end of month, 60 days end of month, etc)
- Billing number
- Supplier : the supplier with whom the order is placed. Only the catalogue references from this supplier may be added to the order.
- Place of delivery of the order : indicates the place where the equipment will be delivered. This option has value only in the entity in which the order is created.
- Description : description of the order
- Status : Indicates the current status of the purchase order in the release run.
- Estimated delivery date : indicates the indicative date by which the supplier must deliver the order.

It is also possible to :

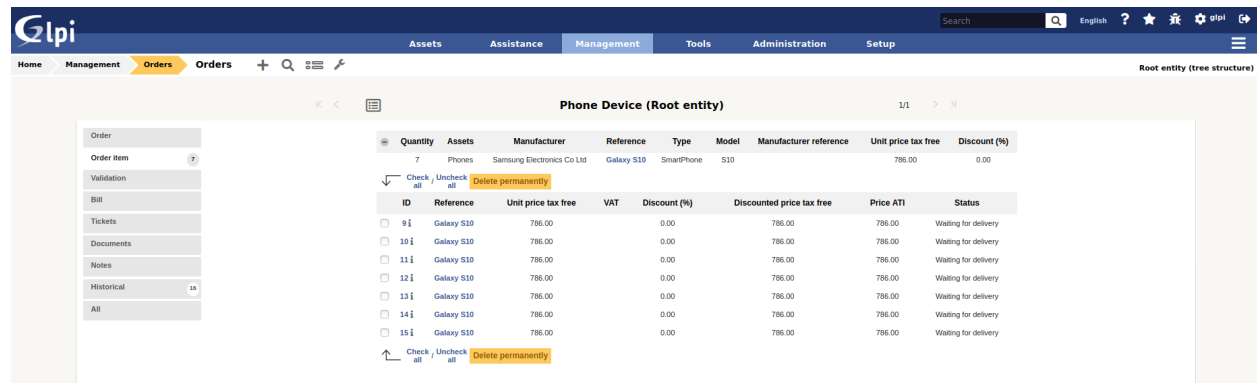
- Define author and recipient

## 11.6.1 Adding articles

The screenshot shows the GLPI interface for managing purchase orders. The top navigation bar includes 'Assets', 'Assistance', 'Management', 'Tools', 'Administration', and 'Setup'. The 'Management' tab is active, and the 'Orders' sub-tab is selected. The main content area displays the 'Phone Device (Root entity)' page. On the left, a sidebar menu lists various order-related actions: Order, Order item, Validation, Bill, Tickets, Documents, Notes, Historical, and All. The 'Details' tab is active, showing a form to 'Add to the order from the catalog'. The form includes fields for 'Type' (Phones), 'Product reference' (Galaxy S10 - A5413TG9484), 'Quantity' (1), 'Unit price tax free' (786), 'VAT' (No VAT), and 'Discount (%)' (0). An 'Add' button is visible at the bottom right of the form. The footer of the page contains copyright information for GLPI 9.4.5 and FusionInventory 9.4+2.3.

In the Details tab, it is now possible to add products to the order.

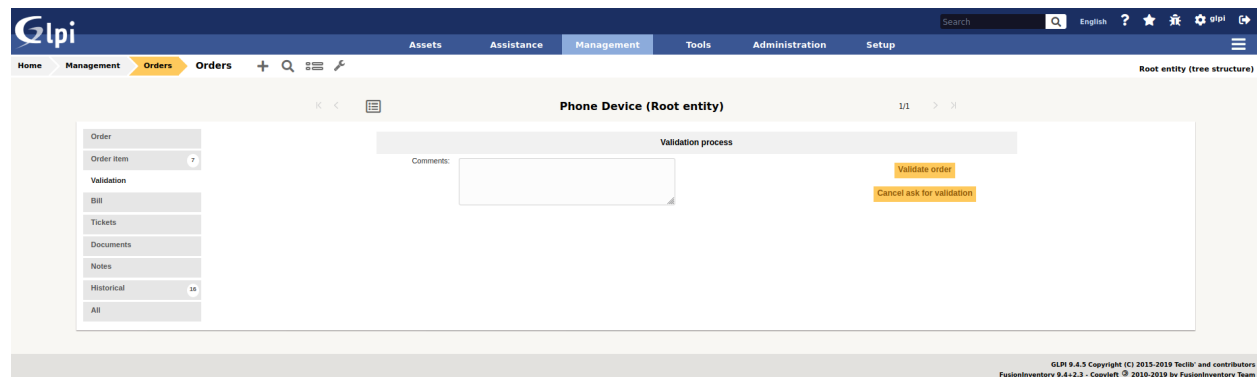
- Select the type of material to add to the order
- Select product number
- Indicate the quantity to order
- Change the unit price if it is not the price indicated in the product reference.
- Indicate the percentage discount on products
- You can then validate the addition of these products to the order.



Quantity	Assets	Manufacturer	Reference	Type	Model	Manufacturer reference	Unit price tax free	Discount (%)
7	Phones	Samsung Electronics Co Ltd	Galaxy S10	SmartPhone	S10		786.00	0.00
9	Galaxy S10						786.00	0.00
10	Galaxy S10						786.00	0.00
11	Galaxy S10						786.00	0.00
12	Galaxy S10						786.00	0.00
13	Galaxy S10						786.00	0.00
14	Galaxy S10						786.00	0.00
15	Galaxy S10						786.00	0.00

Note : it is possible to modify the price and the discount of a set of materials when the order is being edited. To do this, simply double-click on one of these 2 fields, modify its value and then validate the change.

## 11.6.2 Validation workflow



The plugin has a validation workflow that works as follows :

- Creating the order : status In progress
- Request for order release : status Pending approval
- Order confirmed : the order then goes to the status In process of delivery.
- Order completely received : automatic switch to Delivered status
- Cancelled purchase order : Change to status Cancelled

The definition of the persons having the right to make a validation request, to validate or to cancel an order is done in the edition of the profiles.

## 11.6.3 Receiving the order

Once the order has been confirmed, the equipment can begin to be received. To do this, you must go to the Receiving Equipment(s) tab. Each product is identified by a line with the following information :

- Product reference
- Number of equipment(s) received
- Number of material(s) related to inventory items in GLPI
- Unit price of the product (excl. VAT)
- Unit price of the product (incl. VAT)
- Price per unit awarded (excl. VAT)

By clicking on the + icon on the left of the line, it is possible to access the details of the products ordered.

The screenshot shows the GLPI interface with the 'Management' tab selected. The 'Orders' section is active, displaying a list of ordered items for 'Phone Device (Root entity)'. The table has columns: ID, Reference, Status, Delivery date, Delivery form, and Delivery status. The status for all items is 'Waiting for delivery'.

ID	Reference	Status	Delivery date	Delivery form	Delivery status
9	Galaxy S10	Waiting for delivery			
10	Galaxy S10	Waiting for delivery			
11	Galaxy S10	Waiting for delivery			
12	Galaxy S10	Waiting for delivery			
13	Galaxy S10	Waiting for delivery			
14	Galaxy S10	Waiting for delivery			
15	Galaxy S10	Waiting for delivery			

## 11.6.4 Receiving material

It is possible to receive the material that has been ordered, from the moment the order is validated. You have to go to the tab Reception of material(s). One line is displayed for each product reference ordered. This line contains the following information :

- Product reference
- Received equipment(s) : indicates the number of equipment received out of the total number of equipment ordered for this product.
- Related equipment(s) : number of goods received related to inventory items in GLPI
- Price per unit (excl. VAT)
- Unit price (incl. VAT)
- Price per unit awarded (excl. VAT)

In order to proceed to the reception of products, just click on the + icon on the left of the line.

There are 2 ways to receive material :

- Reception of materials selected from the list of products ordered but not yet received
- Receipt of materials in bulk : in this case we will indicate how many products are received (without first selecting from the list).

The screenshot shows the GLPI interface with the 'Management' tab selected. The 'Orders' section is active, displaying a list of ordered items for 'Phone Device (Root entity)'. An 'Actions' dialog box is open, showing the 'Take item delivery' action. The dialog box contains fields for 'Delivery date' (2020-01-29), 'Delivery form', and 'Delivery status' (10 Post).

In both cases, it is necessary to return the delivery note associated with the reception of the material. Alternatively, a status of the received objects can be indicated (e.g. Conforming, Non-conforming, etc). Once received, it is possible to :

- Generate a material in GLPI from the received product
- Link the product received to existing GLPI equipment

## 11.6.5 Generate the material

The screenshot shows the GLPI interface with the 'Phone Device (Root entity)' selected. A modal window titled 'Generate item' is open, displaying a table with the following columns: Product reference, Name, Serial number, Inventory number, Template name, Entity, Location, Group, and Status. The table contains several rows of data for 'Galaxy S10' products. The 'Generate item' button is located at the bottom right of the modal.

In order to generate materials from received products, simply select from the list those for which this operation is to be performed. Then select the Generate associated material option. A screen will appear. Each line represents a received product, and the following information must be given :

- Name of the material generated in GLPI
- Serial code
- Asset number

Note :

- If there is a template associated with the reference of the generated product, and if the latter is created in the same entity as the reference, then the inventory object in GLPI will be created from this template.
- The financial information of the order will be carried in the newly created inventory object.

## 11.6.6 Link to an inventory item

The screenshot shows the GLPI interface with the 'Phone Device (Root entity)' selected. A modal window titled 'Link to an existing item' is open, displaying a table with the following columns: ID, Reference, Status, Delivery date, Associated items, and Serial number. The table contains one row of data for 'Galaxy S10' with status 'Taken delivery' and delivery date '2020-01-29'. The 'Link to an existing item' button is located at the bottom right of the modal.

It is possible not to generate a received product, but to link it to material already present in GLPI. To do this, you must select the option Link to an existing material. A drop-down list will appear and will present all the materials present in GLPI which :

- Are in the entity of the order (or in a sub-entity if the order is visible in the sub-entities)
- With the same type of equipment
- With the same type of equipment
- Identical

Once linked to a product, the material in GLPI will receive the financial information from the order.

## 11.7 Rights management

The screenshot shows the GLPI Administration interface. The left sidebar contains a menu with items like Profile, Assets, Assistance, Life cycles, Management, Tools, Administration, Setup, Users (1), Historical (10), FusionInventory, File injection, Orders, Tree view, Item's Lifecycle, and All. The main content area is titled 'Super-Admin' and shows a table of permissions for the 'Orders' entity. The table has columns for various actions and rows for different entities. The 'Orders' row has checkmarks in all columns, indicating full permissions. The 'Products references' row has checkmarks for 'Read', 'Update', 'Create', 'Delete', 'Purge', 'Read notes', 'Update notes', 'Order validation', and 'Edit a validated order'. The 'Bills' row has checkmarks for 'Read', 'Update', 'Create', and 'Delete'. The 'Select/unselect all' row has checkmarks for 'Read', 'Update', 'Create', 'Delete', and 'Purge'. A 'Save' button is located below the table.

	Read	Update	Create	Delete	Purge	Read notes	Update notes	Order validation	Edit a validated order	Order Generation	Take item delivery	Cancel order	Generate order without validation	Select/unselect all
Orders	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Products references	✓	✓	✓	✓	✓									✓
Bills	✓	✓	✓	✓	✓									✓
Select/unselect all	✓	✓	✓	✓	✓									✓

Save

GLPI 0.4.5 Copyright (C) 2013-2019 'teclib' and contributors  
FusionInventory 9.4.2.3 - Copyright © 2010-2019 by FusionInventory Team

The plugin rights management allows to define, by profile, if the user has the right to :

- Manage commands
- Manage product references
- Manage invoices
- Confirm an order
- Cancel an order
- Modify a validated order





# CHAPITRE 12

---

## Champs additionnels

---

- Lien des sources : <https://github.com/pluginsGLPI/fields>
- Téléchargement : <https://github.com/pluginsGLPI/fields/releases>

Le plugin champs additionnels vous permet d'ajouter de nouveaux champs à différents objets de GLPI : tickets, ordinateurs, utilisateurs, ...

### 12.1 Pré-requis

Ce plugin requiert :

- GLPI 0.85 ou supérieur
- PHP 5.4 ou supérieur lorsqu'il est utilisé avec GLPI 9.1, et PHP 5.5 ou supérieur lorsqu'il est utilisé avec une version antérieure de GLPI.

### 12.2 Fonctionnalités

- Ajout de blocs conteneurs sur différents objets,
- Ajout de champs dans les blocs,
- Affichage des blocs dans un onglet existant ou dans leur propre onglet,
- Plusieurs types de champs disponibles,
- Les champs peuvent être marqués comme étant requis,
- Gestion des accès des profils par conteneur,
- ...

### 12.3 Installer le plugin

- Décompresser l'archive.
- Déplacer le répertoire `fields` dans le répertoire `<GLPI_ROOT>/plugins/`.
- Aller à la page *Configuration > Plugins*,

- Installer et activer le plugin.

## 12.4 Utilisation

Le plugin va créer les tables requises dans la base de données ainsi que certains fichiers sur le disque (dans <GLPI\_ROOT>/files/\_plugins/fields) automatiquement. Ces fichiers et tables seront mis à jour avec le plugin.

L'utilisation du plugin est relativement simple :

- créer un bloc lié avec des objets,
- créer des champs dans ce bloc.

Vous accéderez à la configuration du plugin via le menu *Configuration > Champs supplémentaires*

### 12.4.1 Ajouter un nouveau bloc

**New item - Block**

Label :

Type :

Associated item type : 

Assets - Computers

Administration - Users

Active :

- *Libellé* : le libellé du bloc qui sera affichée sur les formulaires des objets,
- *Type* : le type d’affichage, parmi :
  - *Ajout d’un onglet* : ajoutera un nouvel onglet sur le formulaire de l’objet qui contient le bloc,
  - *Insertion dans le formulaire* : ajoutera le bloc en bas du formulaire principal de l’objet,
  - *Insertion dans le formulaire d’un onglet spécifique* : ajoutera le bloc en bas d’un onglet spécifique de l’objet.
- *Type d’élément associé* : une liste de types sur lesquels le bloc sera ajouté,
- *Actif* : si le bloc est actif ou non.

---

**Note :** Vous ne pouvez ajouter qu’un seul bloc de type *Insertion dans le formulaire* pour un type d’objet associé.

---

L’ajout d’un bloc dans un onglet spécifique n’est possible que pour un seul objet :

**New item - Block**

Label :

Type :

Associated item type :

Tab :

Active :

Une fois le bloc créé, il est possible de changer son libellé ou son état actif, mais pas son type ni la liste des éléments associés.



## Profils

Pour pouvez affiner les accès des profils en utilisant l'onglet *Profils* du bloc. Vous aurez la possibilité de choisir entre *Pas d'accès*, *Lecture* ou *Écriture* pour chaque profil existant.

Profils	
Self-Service	Write ▲
Observer	No access
Admin	Read
Super-Admin	Write ▼
Hotliner	Write ▼
Technician	Write ▼
Supervisor	Write ▼
Read-Only	Write ▼

Save

### 12.4.2 Ajout d'un nouveau champ

**New item - Field**

Label :

Type :

Default values :

Active :

Mandatory field :

Read only :

Add

[Add a new field](#)

No field for this block

- *Libellé* : le libellé du bloc qui sera affichée sur les formulaires des objets,
- *Type* type du champ, parmi :
  - *Entête* : un intitulé d'en-tête, pour une distinction visuelle,
  - *Texte (ligne simple)* : une ligne de texte simple (`input/@type=text`)
  - *Texte (lignes multiples)* : un champ de texte multilignes (`textarea`)
  - *Nombre* : un nombre (pas de texte autorisé)
  - *Liste déroulante* : une liste déroulante configurable, les valeurs sont paramétrées depuis la configuration des intitulés de GLPI que vous trouverez dans le menu *Configuration > Intitulés*.
  - *Oui/Non* : une liste déroulante avec uniquement les valeurs *Oui* et *Non*,

- *Date* : une date seule avec un sélecteur,
- *Date et heure* : un champ de date avec un sélecteur et une liste déroulante pour l'heure (le pas étant configuré depuis le cœur de GLPI),
- *Utilisateurs* : une liste d'utilisateurs,
- *Actif* : si le champ est actif ou non,
- *Lecture seule* : si le champ est en lecture seule,
- *Valeurs par défaut* : valeurs par défaut du champ,
- *Champ obligatoire* : détermine si le champ est obligatoire.



**Avertissement :** Bien entendu, il vous faudra prêter attention en modifiant les différentes options... Si par exemple, vous définissez un champ obligatoire et en lecture seule, et que vous ne définissez pas de valeur par défaut, la validation du formulaire échouera.

Un autre exemple : si vous définissez une valeur texte par défaut pour un champ nombre... Ce genre de choses ;)

Une fois les champs créés, vous pouvez les réorganiser ou les modifier depuis l'onglet champs du bloc :

Add a new field						
Label	Type	Default values	Mandatory field	Active	Read only	
A field in the block	Text (single line)		Yes	Yes	No	
Does this work?	Yes/No		No	Yes	No	
Which date?	Date		No	Yes	No	
Make your choice	Dropdown		No	Yes	No	
The user	Users		No	Yes	No	
An inactive field	Text (multiples lines)		No	No	No	

Et voyez ce à quoi cela ressemble sur un objet *Utilisateur* par exemple :

User	A field in the block : *	<input type="text"/>	Does this work? :	No ▼
Authorizations 1	Which date? :	<input type="text"/>  	Make your choice :	----- ▼ ⓘ +
Groups	The user :	----- ▼ ⓘ		
Settings		<input type="button" value="Save"/>		
Used items				
Managed items				
Created tickets 1				
Problems				
Changes				
Documents				
Reservations				
Synchronization				
External links				
Historical 3				
Block as tab				
All				

## 12.5 Recherche

Tous les champs qui ont été ajoutés sont disponibles dans les recherches des objets attachés.

## 12.6 Interface simplifiée

Les seuls blocs attachés à des tickets et de type *Insertion dans le formulaire* seront affichés dans l'interface simplifiée. Bien sûr, les droits du profil courant seront également pris en compte !

## 12.7 Traductions

Nouveau dans la version 1.4.0.

Le plugin lui même est traduit par l'intermédiaire des [services de transifex](#) ; mais vous pourrez également traduire les libellés des blocs et des champs ; le processus est identique pour les deux :

Sur le formulaire du bloc, choisissez l'onglet *Traductions* :

Lors de la création d'un bloc, une nouvelle traduction dans la langue courante sera ajoutée avec le libellé renseigné sera créée ; vous pouvez ajouter autant de traductions que vous le souhaitez.

Les valeurs des listes déroulantes peuvent être traduites en utilisant la configuration des intitulés du cœur.

**Avertissement :** Dans GLPI 9.1.1 ; certains problèmes empêchent la traduction des valeurs des listes déroulantes ; la possibilité a donc été désactivée dans le plugin. Elle sera réactivée lorsque le cœur aura été corrigé.

## 12.8 Entités

Si vous utilisez des entités dans votre instance de GLPI, un bloc ne sera affiché que si l'entité à laquelle il est attaché correspond est compatible avec celle de l'objet auquel il s'applique.

Lorsque vous créez ou modifiez un bloc en mode multi-entités, vous devrez choisir s'il sera accessible depuis les entités enfants.

Par exemple, disons que vos entités respectent la structure suivante :

- racine
  - A
    - C
  - B

La table ci-dessous vous dira dans quels cas votre bloc sera affiché ou non :

Entité du bloc	Bloc récursif	Entité de l'objet	Blo affiché
A	Oui ou Non	A	Oui
A	Oui ou Non	B	Non
A	Oui	C	Oui
A	Non	C	Non





- Sources link : <https://github.com/pluginsGLPI/uninstall>
- Download : <https://github.com/pluginsGLPI/uninstall/releases>

### 13.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

### 13.2 Features

This plugin, named « uninstall » or « Item Uninstallation », allows you to remove / replace devices from the inventory by automating certain actions on the fields.

It allows to manage the different stages of the life cycle of a hardware. It has been designed, among other things, to handle the following cases :

- scrapping of equipment
- removal from inventory of temporary equipment due to breakdown
- Replacement of equipment under a warranty for example (See : Replacement Equipment Feature)

Once installed, the plugin includes :

- a rights management directly accessible from Administration -> Profiles (in the Plugins tab -> uninstall),
- a « hardware uninstall » menu in « Configuration »,
- an « uninstall » menu in the user preferences.

### 13.3 Right managment

It is possible to deny access to the plugin or to allow it by specifying the read or write option on the templates.

To do this, simply go to the menu « Administration » > « Profiles », to select the profile you wish to modify, then in the « Uninstallation » tab set the following options :

- *No access* : the plugin does not appear in the menu “Plugin”.
- *Playback* : only the use of previously created templates is possible
- *Writing* : Template creation is possible.
- *Hardware replacement* : standard hardware replacement is possible

## 13.4 Uninstall feature

This functionality of the plugin “Item Uninstallation” adds the possibility of modifying the value of certain fields and of removing the inventory of a computer of the OCS base or the cleaning of the FusionInventory tables.

The uninstall templates can be accessed in the « Administration > Uninstall Hardware » menu.

You can create as many uninstall templates as you want. Each template can contain different features, and be visible in the sub-entities or not.

Uninstall equipment is a transfer of the equipment to itself. During its transfer, a certain number of actions are performed, as specified in the definition of transfer.

- *Name* : the name of the uninstall template
- *Sub-entities* : indicates whether the template is visible in the creation entity and its sub-entities
- *Model type* : indicates which type of model should be used (either Uninstall or Replace)
- *Comments* : free fields to add details about the model
- *Transfer template to be used* : Indicates the transfer template to be used when uninstalling the system.
- *New hardware status* : indicates which status the hardware should take once uninstalled
- *New group* : indicates the group to which the hardware can belong once uninstalled.
- *Software History Deletion* : Allows deletion of all lines in a computer’s history that are relevant to software installations/uninstallations.
- *REMOVE Name* : removes the name of the hardware during uninstallation
- *RAZ Contact* : removes the contact reassembled by OCS during uninstallation
- *Network reset* : resets the « network » field of the equipment record to zero.
- *RAZ OS* : Removes OS information (OS, Version, Service Pack, Product Key, Product ID) when uninstalling.
- *RAZ domain* : removes the « domain » field from the hardware
- *Remove IP & Gateway & Hacker & Subnet* : removes network information (except for the MAC address, which is hardware-related) during uninstallation
- *Remove the machine from OCS* : when uninstalling a computer in GLPI, remove from the OCS base the corresponding machine



— *Delete FusionInventory info* : Delete all FusionInventory info for this material

## 13.5 Uninstall hardware

There are 2 possibilities to uninstall a hardware :

- from its file, in the tab « Uninstallation ».
- from the massive modifications list

Since a template can be recursive (visible in the sub-entities), the list of templates displayed in the massive actions indicates all the templates available in the current entity (either those defined in this entity, or in an entity above it and visible in the sub-entities).

### 13.5.1 Change of location on uninstallation

It is possible to change the location of the hardware once it has been uninstalled. This selection is made either :

- in the hardware file, once the uninstallation model has been chosen.
- from the user's preferences for uninstallation by massive modification

It should be noted that a location is declared within an entity. It is therefore possible, depending on the current entity, to define an uninstall location.

## 13.6 Replace feature

This feature of the “Item Uninstallation” plugin adds the possibility to replace one hardware by another.

### 13.6.1 Replacement Model Form

Replacement models are managed in the same way as uninstall models : They are accessible in the menu « Administration > Uninstall hardware ». Each model can contain different characteristics, and be visible in the sub-entities or not.

In this form there are 4 tabs : \* Main : allows to display on the page only the main information of the template. \* Replacing data : displays the various actions to be performed when replacing one piece of equipment with another. \* All : displays all the tabs on the same page

GLPI 9.4.5 Copyright (C) 2015-2019 Teclib and contributors  
FusionInventory 9.4+2.3 - Copyright © 2010-2019 by FusionInventory Team

Here is the description of the fields of the main tab :

*Name* : the name of the replacement model  
*Sub-entities* : indicates whether the template is visible in the creation entity and its sub-entities  
*Model type* : indicates what type of model should be used (here we are talking about type - Replacement)  
*Comments* : free fields to add details about the model  
*New hardware status* : indicates which status the old hardware should take once it has been replaced

## 13.6.2 Replacing data tab

GLPI 9.4.5 Copyright (C) 2015-2019 Teclib and contributors  
FusionInventory 9.4+2.3 - Copyright © 2010-2019 by FusionInventory Team

Here is the description of the fields of the actions tab.

### General Information

- *Copy name* : allows you to define if you want to copy the name of the old hardware to the new one.
- *Copy the serial number* : Allows you to define if you wish to copy the serial number of the old equipment to the new one.
- *Copy the inventory number* : allows you to define if you want to copy the inventory number of the old equipment to the new one.
- *Overwrite information* (from the old material to the new one) : this option allows you to choose the behavior to adopt if one of the unique information (Reservation, Financial Information, Name, Serial, Otherserial, Entity...) is present in the old material AND in the new material : *Yes* : by selecting YES, the information in the

old material will replace that in the new material. *No* : by selecting NO, the information of the new hardware will NEVER be overwritten by the information of the old hardware.

#### *Method of archiving old material*

In this option, we define the behavior that the uninstall plugin must adopt with the old hardware :

- *Purge + PDF / CSV* : if you have the PDF plugin the old hardware will be purged from the GLPI database and its information will be attached to the new hardware in a PDF document. If you do not have the PDF plugin, they will be stored in CSV format.
- *Deletion + Comments* : the old hardware is put in the trash and a comment is added in each of the two hardware (« This hardware has been replaced by... » and « This hardware has replaced the hardware... »).

#### *Data location - Connection with other hardware*

- *Copy Documents* : Allows you to attach documents from old equipment to new equipment.
- *Copy Contracts* : allows you to attach the contracts from the old equipment to the new equipment.
- *Copy financial information* : allows you to attach financial information from the old equipment to the new equipment. Warning : if the « Overwrite information » option is activated, even if the new hardware has financial information, it will be replaced by the old hardware.
- *Copy Reservations* : Allows you to attach the reservations made on the old equipment to the new equipment. Attention : same remark as for the financial information.
- *Copy the user* : allows you to pass on the user of the old equipment to the new one. Caution : same remark as for the financial information.
- *Copy group* : Allows you to assign the group of the old equipment to the new equipment. Caution : same remark as for the financial information.
- *Copy the tickets* : Allows you to attach the tickets of the old equipment to the new equipment.
- *Copy network connections* : Allows you to transfer the network connections from the old hardware to the new hardware.
- *Copy direct connections* : Allows you to transfer the direct connections (Monitor, Printer, Device, Phone) from the old hardware to the new hardware.

## 13.7 Replace hardware

As well as uninstalling equipment, there are two ways to replace equipment :

- from the card of this one, in the tab « Plugins -> Uninstallation »,
- from the massive modification list.

Since a replacement template can be recursive (visible in the sub-entities), the list of templates displayed in the massive actions indicates all the templates available in the current entity (either those defined in this entity, or in an entity above it and visible in the sub-entities).

Once you have selected the hardware to be replaced via the massive actions, or you use the « Uninstall » tab directly in it :

- a summary of the selected template appears in order to show you the actions that will be performed by the plugin.
- the list of the selected hardware(s) appears and you can choose the replacement hardware in the same way as when you choose a hardware in a Support Ticket.

The screenshot shows the GLPI Administration interface. The top navigation bar includes links for Assets, Assistance, Management, Tools, Administration (active), and Setup. The main content area is titled 'Reminder of the replacement model - General informations' and contains several configuration options:

- Copy Name:** Yes
- Copy Serial number:** Yes
- Copy Inventory number:** Yes
- Overwrite informations (from old item to the new):** No
- Archiving method of the old material:** Delete + Comment
- New location of item:** Empty location
- New status of the computer:** Status

Below this, there is a section titled 'Reminder of the replacement model - Connections with other materials' with the following options:

- Copy Documents:** Yes
- Copy Financial and administratives information:** Yes
- Copy Contract:** Yes
- Copy Reservations:** Yes
- Copy User:** Yes
- Copy Group:** Yes
- Copy Tickets:** Yes
- Copy Connections Networks:** Yes
- Copy Direct connections:** Yes

At the bottom, there is a section titled 'Choices for item to replace' with a table:

Old item	Inventory number	Serial number	New item
N/A	Asset: 1234567890	140424546501596	<input type="text"/> i

A 'Replace' button is located below the table. The footer of the page contains the GLPI version and copyright information.

## 13.7.1 Change of location on replacement

It is possible to change the location of the equipment once it has been replaced. This selection is made either : \* in the equipment sheet, once the replacement model has been chosen, \* from the user's preferences for massive modification replacement.

Note that a location is declared within an entity. It is therefore possible, depending on the current entity, to define a replacement location.



- Sources link : <https://github.com/pluginsGLPI/formcreator>
- Download : <https://github.com/pluginsGLPI/formcreator/releases>

FormCreator is a plugin which allow creation of custom forms of easy access.

### 14.1 Features

- Direct access to forms self-service interface in main menu,
- Highlighting forms in homepages,
- Access to forms controlled : public access, identified user access, restricted access to some profiles,
- Simple and customizable forms,
- Forms organized by categories, entities and languages,
- Questions of any type of presentation : Textareas, lists, LDAP, files, etc,
- Questions organised in sections. Choice of the display order,
- Possibility to display a question based on certain criteria (response to a further question),
- A sharp control on responses from forms : text, numbers, size of fields, email, mandatory fields, regular expressions, etc,
- Creation of one or more tickets from form answers,
- Adding a description per fields, per sections, per forms, entities or languages,
- Formatting the ticket set : answers to questions displayed, tickets templates,
- Preview form created directly in the configuration,
- Translation of forms to serve them in several languages.

You can take a look to [the full changelog from plugin sources](#).



### 14.2 Install the Plugin

- Uncompress the archive.

- Move the `formcreator` directory to the `<GLPI_ROOT>/plugins` directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

**Avertissement :** The plugin's directory must have the same name as the plugin :

- **Good :** `glpi/plugins/formcreator`
- **Bad :** `glpi/plugins/formcreator-master`
- **Bad :** `glpi/plugins/formcreator-0.90-1.3.2`

Only one directory must contains the plugin's files of a single plugin in the GLPI plugins directory. **Don't rename the plugin's directory for backup, move it !**



## 14.3 Service catalog

The service catalog is a replacement for the GLPI's simplified interface.

It is enabled on a per entity basis and exists in two flavors

- simplified
- extended

To enable it, edit an entity, open the *Forms* tab and set the field *Helpdesk mode* to *Service catalog simplified* or *Service catalog extended*. This setting handles inheritance from parent entity to children entities.

Users using the simplified interface will benefit a new interface allowing them to :

- browse forms and FAQ with the unified interface
- follow the process of their requests
- book assets
- view their feeds

Users using the extended interface have a more complete view on their requests.

Forms with *Direct access on homepage* enabled will appear in the interface. Users may search by browsing the categories on the left of the screen, and may also search for forms with a natural language search engine.

By default FAQ items in the Knowledge Base also show with forms. The only prerequisite is to associate form categories to knowledge base categories (in *Setup > Dropdowns : Forms > Form categories*).

It is possible to have a distinct display of FAQ in *Administration > Entities > Forms*. When **Distinct menu entry** is selected the service catalog shows a new menu entry on the left. FAQ are browsable by category and key words.

---

**Note :** All entity settings for Formcreator propagate to sub entities by default. The administrator may change the settings in a sub entities to break inheritance.

---

### 14.3.1 Overview

The service catalog shows several counters on the top left corner. Those counters show all requests involving the current user as a author, requester, observer, assigned or validator.

The requests displayed here are

- tickets created without Formcreator
- tickets generated by forms having only one target tickets
- form answers when the form has no target ticket

- form answers when the form has several target tickets
- The request are spread over 4 counters
- processing
  - pending
  - to validate
  - closed

---

**Note :** New requests are not included in the counters.

---

### 14.3.2 RSS feeds

When users are allowed to view RSS feeds, the service catalog shows a menu entry on the left to access them. RSS feeds administration is done in GLPI as usual.

### 14.3.3 Reservations

Users may book an asset with GLPI's reservation system, available in the left menu of the service catalog.



## 14.4 Forms configuration

### 14.4.1 Form creation

---

**Note :** The right to create forms is enabled to the profiles having the right to update entities in GLPI. Refer to the documentation of GLPI to know how to setup this right.

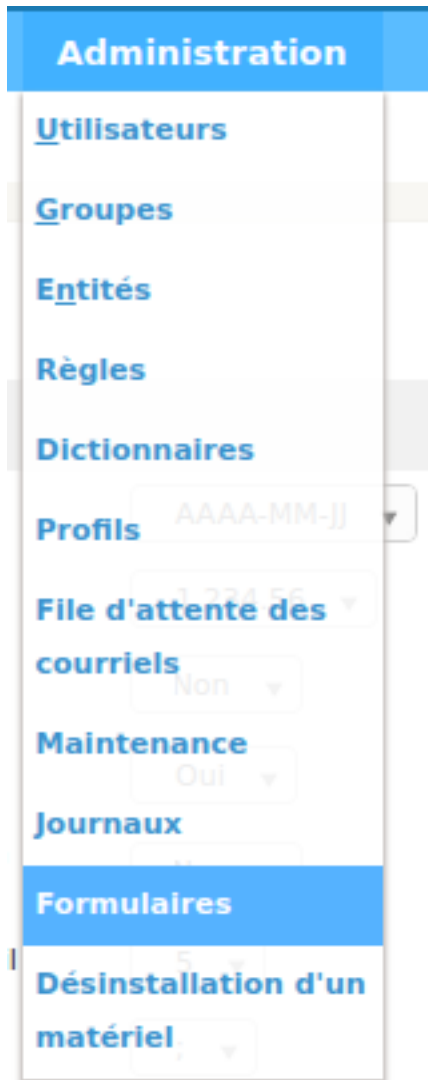
---

---

**Note :** Forms must be created in the entities where they are intended to be available. A form may be available in its entity and all sub entities if the field **Child entities** is set to **Yes**.

---

1. Navigate to **Administration > Forms**.



2. Click on the button to add a form +

Following fields must be populated :

- **Name** : Name of the form.
- **Active** : A form is inactive by default. You need to explicitly activate it when it is ready to use.

It is recommended to fill the Category field :

- In the simplified interface, categories allow to tidy forms by Blocks.
- If you want to use the **service catalog** you must use form categories.

---

**Note** : Form categories are plugin's dropdowns. You can add form categories directly when editing a form using the + next to category field. You can do the same from **Configuration > Dropdowns > Form category**.

---

- **Direct access on homepage** : Direct access to the form from the GLPI's simplified interface.
- **Description** : displays in the list forms.
- **Language** : by default a form is set to the language of its creator. The form will be available only to users using the same language as the form. Choose **All languages** to make the form available to users without language restriction.
- **Header** : displays when the form is displayed.
- **Need to be validate** : If **Yes** a list of validators is displayed. This is a list of GLPI users (with the right **Validate an incident** or **Validate a request** (in its profile) on an compatible entity with the form's entity. The list of



validators is a multiple choice list. If no validator is selected all of them are submitted when the form is being used.

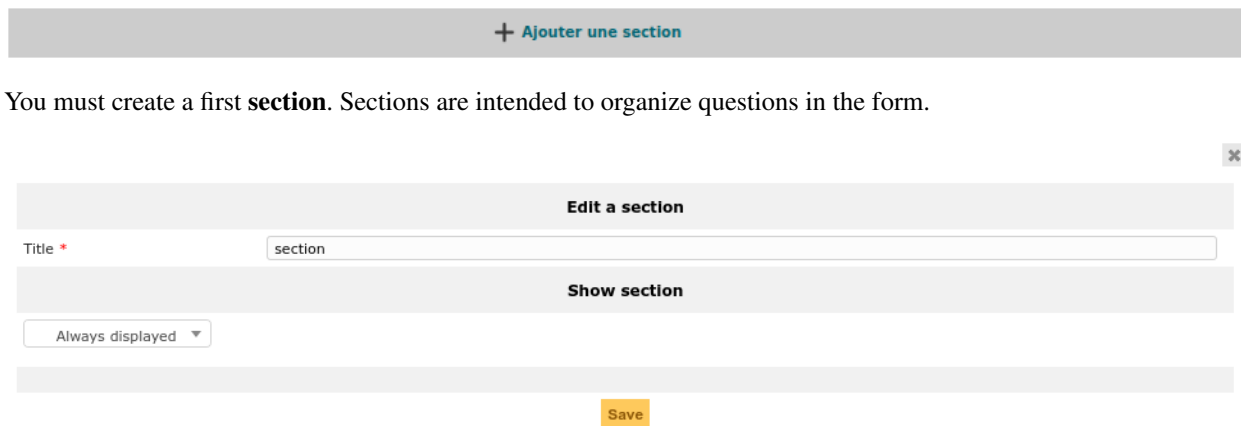
- **Default form in service catalog** : if **Yes** the form will display in the service catalog without being filtered by the current category or keywords.

When all fields are filled, click on the **add button** at the bottom of the page.

**Note** : Deleting a form is possible only if there are no associated answers. To delete a form, delete all its answers first from the **Form answers** tab.

## 14.4.2 Sections

After clicking on the tab **Question** the following page displays :



The screenshot shows a form titled "Edit a section". It contains a text input field for the title, currently containing the word "section". Below the input field is a button labeled "Show section". Underneath the button is a dropdown menu with the text "Always displayed" and a downward arrow. At the bottom of the form is a yellow "Save" button.

It is possible to setup conditions to show the whole section, based on the answers provided to questions located in other sections of the form.

## 14.4.3 Questions

Click on the link **Add a question** in a section of your choice.



The screenshot shows a header for "Section 1". Below the header, there is a link that says "+ Ajouter une question", which is highlighted with a red rectangular box. Below this link is another link that says "+ Ajouter une section".

The following page displays :

A question is made of :

- a title : this is the label of the question in the form.
- a type : see list below.
- a section : the section containing the question. You can move a question to an other section with this field.
- a description ; it is displayed under the question in the form. Use it as a hint for the requester, telling him which content is expected.
- a dropdown list **Show field** to enable a condition

It submits the following choices :

- **Always visible** : the field is always displays
- **Hidden unless** : The question is hidden except if answers to other questions matches a condition.
- **Displayed unless** : The question is displayed except if the answers to other questions matches a condition.

Conditions may be multiple. To add or remove a condition two buttons are available :

#### Note : List of pictograms

- The **circle** allows you to make a question mandatory



, or optional



- **Arrow up** and **arrow down** allow you to reorder questions in a section.
- Clicking on a question allow you to edit it.
- **Two stacked squares** allows you to duplicate a question or a whole section.
- The **recycle bin** allows you to delete a question or a whole section.

Questions are organized on a 4 columns array. You can resize width of questions, reorder them using drag and drop and put up to 4 questions on the same row.

#### 14.4.4 Types of question

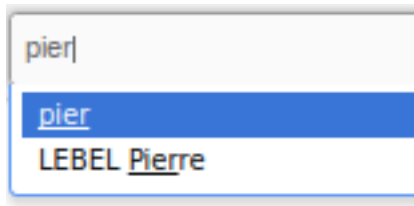
There are about twenty types of questions available. Depending on the chosen type, you need to provide additional informations.

---

**Note :**

- **Required : Yes/No.** When running the form a red star shows next to label of questions requiring an input.
  - **Default values :** its content depends on the type of the question.
  - **Range Min/Max :** Restricts the value to the given range when running the form.
  - **Additional validation (Regular expression) :** You may set a custom regex with a regular expression. Use it when other restriction methods cannot satisfy your needs. Don't forget to specify the delimiters of the regex. You may add any modifier after the closing delimiter.
- 

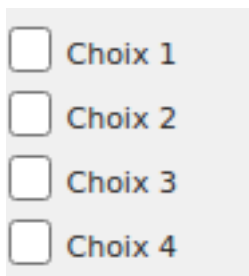
#### Actors

A screenshot of a web form element. It shows a text input field with the text 'pier|' inside. Below the input field is a dropdown menu that is open, showing two options: 'pier' (highlighted in blue) and 'LEBEL Pierre'.

This field allows you to choose one or several users :

- in GLPI, available in **Administration > Users**
- not in GLPI, by typing an email address

#### Checkboxes (multiple choices)

A screenshot of a web form element showing a list of four checkboxes. Each checkbox is followed by a label: 'Choix 1', 'Choix 2', 'Choix 3', and 'Choix 4'. All checkboxes are currently unchecked.

### Modifier une question

**Titre\***  **Type\***  Boîtes à cocher ▾

**Section\*** Section 1 ▾

**Champ obligatoire** .Non ▾

**Valeur(s) par défaut(Une par ligne pour les listes)**  **Valeurs(Une par ligne)**

**Taille** Min  Max

**Description**



**Affichage du champ**

Toujours visible ▾

**Enregistrer**

Add a line per value in **Values**. Default values may be set in **Default values**, one per line. If this field is empty, no checkbox will be ticked.

## Date / Datetime / Time

🕒

Avril ▼

2017 ▼

🕒

Sem.	L	M	M	J	V	S	D
13	27	28	29	30	31	1	2
14	3	4	5	6	7	8	9
15	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23
17	24	25	26	27	28	29	30

Heure 00:00

Heures 00 ▼

Minutes 00 ▼

Maintenant

Terminé

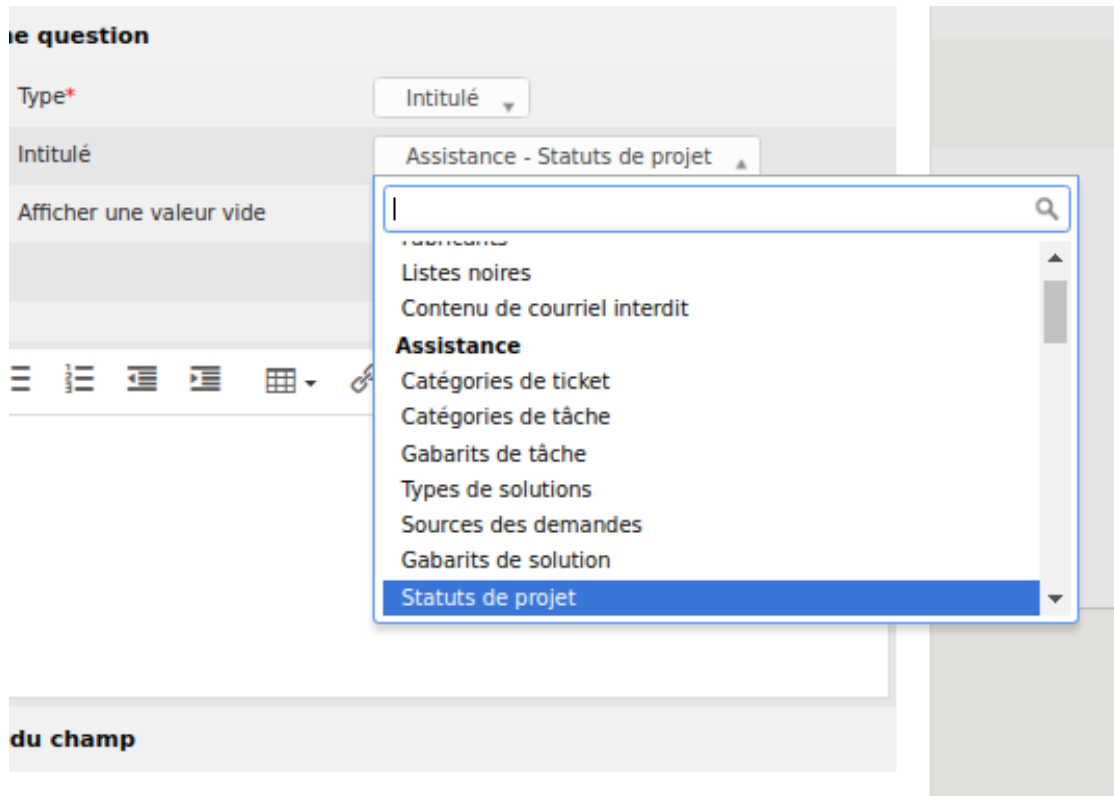
This field allows to select a date, a datetime or a time from a mini calendar.

## Description

This field only displays informations. Use it to give more details about a question.

## Dropdown

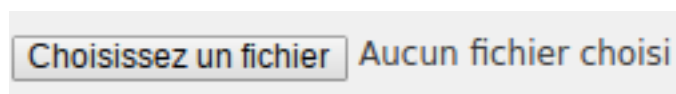
This field allows the user to choose a value among those available in a dropdown from GLPI (in **Configuration > Dropdowns**).



## Email

The answer to this type of field must be a syntactically valid email address.

## File



This field allows the requester to upload a file.

## Float

This field must be an float value. Note you may use a regular expression to tighter restrict the answer.

### GLPI Object

This field allows you to build a dropdown from a GLPI object among those available in the menus :

- Assets
- Assistance
- Management
- Tools
- Administration

### Hidden field

This field is hidden and allows to retrieve an arbitrary value when generating the target (ticket or change).

---

**Note :** This type of questions never shows on a form.

---

### Hostname

This field is invisible. It allows to get the hostname if the computer used by the requester, assuming the DNS is able to properly solve it from its IP address

### Integer

This field must be an integer value. Note you may use a regular expression to tighter restrict the answer.

### IP Address

This field is hidden and collects the IP address of the form requester. It does not show in the form.

### LDAP select

This field allows you to create a dropdown list with objects from a LDAP directory :

**Note :** This type of questions cannot be used to fill actors of a target.

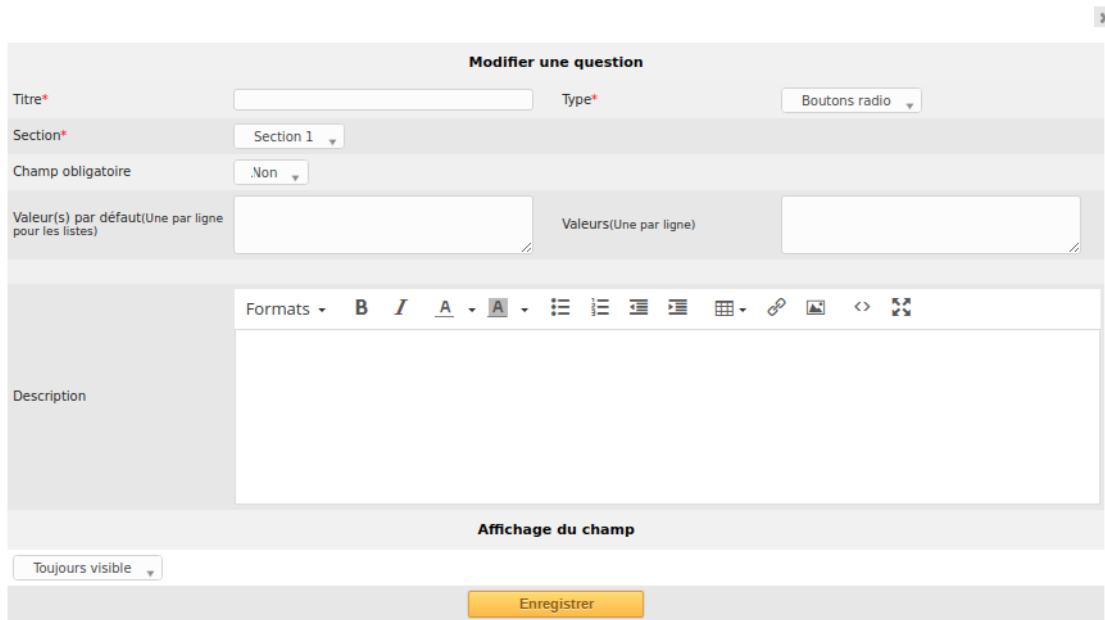
### Multiselect (multiple choice)

This field allows you to create a dropdown list with ability to select multiple items. Values are added one per line in the field **Values**. You may set default values, one per line in *Default values\**. If no default value is set then no item will be selected by default.

### Request type

This field is a dropdown with « Incident » or « Demand » choices. It can be used to set the request type of a ticket being generated by the form.

### Radio buttons (one choice only)



The screenshot shows the 'Modifier une question' (Edit question) interface. It features several input fields: 'Titre\*' (Title), 'Section\*' (set to 'Section 1'), 'Type\*' (set to 'Boutons radio'), and 'Champ obligatoire' (set to 'Non'). There are also input fields for 'Valeur(s) par défaut' and 'Valeurs'. A large 'Description' field with a rich text editor toolbar is present. At the bottom, there is a 'Toujours visible' checkbox and an 'Enregistrer' (Save) button.

Add a line per value in **Values**. A default value may be set in **Default value**. If this field is empty, no radio will be selected.

### Select (one choice only)

This field allows you to create a dropdown list and set its items. Items are added one per line in **Values**. Default value may be set in **Default value**. If there is no default, no item is selected by default.

### Tags

This is an hidden field to add a tag to the form for future processing.

---

**Note :** This type of field is only available when the plugin **Tag** is installed and enabled.

---

### Text

This field allows you to input a single line of text.

### Text area

This field allows you to input several lines of text.

### Urgency

This field allows you to select an urgency defined in GLPI.



### 14.4.5 Conditions

Conditions are expressions involving questions and sections which are evaluated in order to determine

- if a question is displayed
- if a section is displayed
- if the submit button of a form is displayed
- if a target (see below) must be generated

Available operators in an assertion are :

- equals
- not equals
- less than
- greater than
- less than or equal
- greater than or equal
- is visible
- is not visible
- matches a regular expression

The condition expression can be used with the following rules :

- **Hidden unless** : the item is hidden or not generated except when the expression is true
- **Displayed unless** : the item is displayed or generated except when the expression is true

### 14.4.6 Access types

Three values are available :

- **Public access** : Anonymous users may access the form ; you may use it in an intranet.
- **Private access** : Users having a GLPI account may access the form.
- **Restricted access** : Only users having the specified profiles may access the form.

When a form is set to **Public access** it is possible to enable a simple captcha. This captcha should prevent spamming if a form is accessible from internet.

---

**Note :** Captchas are not enabled by default on anonymous forms. Administrators need to enabled them explicitly depending on how the form may be accessible from an hostile network such Internet.

---

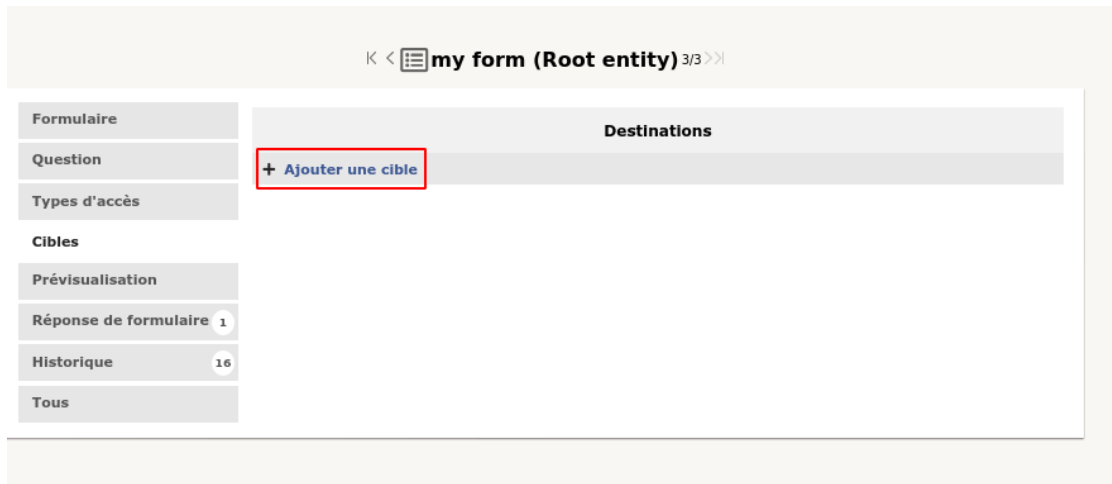
### 14.4.7 Targets

There are two types of targets for a form :

- tickets
- changes

It is possible to generate any number of targets from a single form. Mixing types of targets is also possible.

To create a target click on the link **Add a target**



The following window is then displayed :

Choose a target name and type, then validate. The new target is created and is available for tuning.

### 14.4.8 Preview

This tab allows you to view how the form will be rendered and test it without activating it.

---

**Note :** Submitting answers from the preview will be actually saved, and will generate targets if the form is not configured with validation.

---

### 14.4.9 Form answers

This tab shows all answers saved for the form.

---

**Note :** To delete a form, all its answers must be deleted first. A warning shows at the bottom of the main tab of a form as a reminder.

---

### 14.4.10 Categories

**Menu :** *Setup > Dropdowns : Forms > Form categories*

Form categories allow you to arrange your forms list. Forms are displayed when they belong to the selected category or any sub category. When a form does not have any category, it is displayed when no category is selected or when the user selects « view all ».

You can add or edit categories generally from the Setup menu : *Setup > Dropdowns*.

You can also add new categories directly from the form page like all GLPI dropdowns.

They are defined by entities and can be translated since GLPI 0.85 like all other dropdowns.

**Note :** Dropdowns translation must be enabled on GLPI general configuration page *Setup > General, General setup > Translate dropdowns = Yes*

**Note :** Categories may be associated to *Knowledge base categories*. This link is necessary to allow FAQ entries to show along your forms.

## Configuration

The screenshot shows the GLPI web interface for configuring form categories. The top navigation bar includes the GLPI logo, a search bar, and icons for star, settings, and power. The breadcrumb trail is: Home > Setup > Dropdowns > Form categories. The page title is 'Form categories'. The 'Root entity (Arborescence)' is set to 'Super-Admin'. A dropdown menu labeled 'Dropdowns' is set to 'Forms - Form categories'. Below this, there is a search bar with 'Items seen' and 'contains' filters, a 'Search' button, and a star icon. The 'Display (number of items)' is set to '20'. There is a 'Current page in landscape PDF' button and a 'From 1 to 2 on 2' indicator. An 'Actions' button is visible. The main content area shows a table with a header 'Name' and two rows: 'Computer problems' and 'Human resources'.

## Render

**Welcome to our FormCreator users page.**

You can **select a specific** form below to [easily create a ticket](#) on a specific subject...

**Forms without category**

- + General request

**Computer problems**

- + Network problem
- + Software request

**Human resources**

- + New user

**My last forms (requester)**

✓ Software request	2016-01-06 15:29
✓ General request	2016-01-06 10:01
✓ Network problem	2015-12-18 12:01
✓ Network problem	2015-12-17 12:41
✓ Network problem	2015-12-17 12:38

**All my forms (requester)**

**My last forms (validator)**

## 14.4.11 Questions

After the creation of a form, create fields for for the user to fill out.

**Add a question**

Title\* question Section\* S

Type\* Text

Required No

Default values

Additional validation (Regular expression)

Range Min Max

Formats B I A A List List List List Table Link Image Code Fullscreen

Description

**Show field**

Always displayed

Add

The name of the questions will appear on the left and the field type selected on the right.

The Description will be under the input field.

Additional options may be displayed depending on the currently selected question type.

If validation of the input is desired, it can be implemented following [PHP Regular Expressions](#).

If you want to show or hide questions depending on the answers of other questions, use the *show fields* area when editing a question. In the version 2.5.0 you may use more complex expressions checking for the content of several questions, and use logic operator **OR** and **AND**. The precedence of boolean operators applies, meaning that **AND** has precedence over **OR**.

## 14.4.12 Translation

In some cases a form should be available in several languages. Choose first in which language a form should be created. This language should be english (US or UK) or the language that most of target users understand. This is the fallback language if no alternative is found.

This language is also used as reference in the translation process. Then be sure that the choosen language is readable by the user who will translate the form.

To translate a form open the tab **Form languages**, then create all languages you need to provide to users.

The screenshot displays the GLPI Administration interface. The top navigation bar includes tabs for Assets, Assistance, Management, Tools, Plugins, Administration (selected), and Setup. The left sidebar shows a list of items: Form, Questions, Access types, Targets, Preview, Form answers, Form languages (selected), Historical, and All. The main content area is titled 'translatable form (Root entity)' and shows a 'New Item - Form language' form. The form has a 'Name' field set to 'English', a 'Comment' text area, and a list of existing languages: Language, Deutsch, Français, and 香港. There are buttons for 'Add a new language', '+ Add', and 'Actions'. The bottom of the page shows 'AJAX DEBUG' and 'Display only tab for debug'.

Choose a language added to the form and click it to begin or resume translation, then select the tab **Translations**. Click the button **New translation** and a popup dialog will open and show a string to translate. Type the translation, and click save or validate with **Enter** on your keyboard.

If an other string must be translated, it will show immediately. If no more string needs to be translated, a message will show instead.

When you close the dialog the list of translated strings is refreshed. You can edit a translation by clicking on it, delete one or several translations with the checkboxes on the left of the list and the **Delete** button\*\*. You can also filter the list with the filter input box.



## 14.5 Targets

Targets are objects generated by FormCreator submission. If a form requires validation, the targets are delayed until approbation.

**Note :** Targets are currently tickets only.

A target ticket defines the ticket generated by the form. You may have several targets per form.

Open the tab *Destination* and create a target with a name and a type (currently ticket only).

### 14.5.1 Target ticket

A target ticket generates a ticket.

The ticket is build from scratch or from a ticket template available in GLPI. You may

- customize the name of the target ticket to distinguish it from possible other targets,
- customize the title and description of the ticket using questions and answers,
- set the destination entity of the generated ticket,
- set a time to resolve (formerly due date),
- set the type of ticket (request or incident),
- set the associated elements,
- set the category,
- set the urgency,
- set the location,
- assign tags to the ticket if the plugin *Tags* is available,
- link the generated ticket to other tickets
- define actors of the ticket.
- define conditions to meet to generate the target ticket

**Note :** Setting an urgency, a category or a location (specific or from a question) overrides the urgency defined in a ticket template (if any)

### Destination entity

Many choices are available :

- Current active entity : the entity of the requester who fills a form,
- Default requester's entity : the default entity defined for the user who fills the form,
- First dynamic requester's entity (alphabetical) : sorts all dynamic entities of the requester alphabetically and selects the first one,
- Last dynamic requester's entity (alphabetical) : sorts all dynamic entities of the requester alphabetically and selects the last one,
- The form's entity : the entity where is located the form being filled by the requester,
- Default entity of the validator : the default entity of the validator who validates the request,
- Specific entity : an entity set by the form's designer,
- Default entity of a user type question answer : the default entity of a user chosen by the requester in a question.
- From a GLPI object > Entity type question answer : the entity chosen by the requester from a question.

---

**Note :** Default requester's entity may be not set, then there is a risk to generate tickets in the root entity.

---

---

**Note :** Default entity of the validator may be not set, then there is a risk to generate tickets in the root entity.

---

---

**Note :** Dynamic requester's entity requires user synchronization, then there is a risk to generate tickets in the root entity.

---

### Ticket template

Sets predefined settings for the ticket from a template. All settings defined in the target ticket will override the values found in the template.

### Request type

Sets the type of the generated ticket

- Default or from a template : value set by GLPI when no type is set, or from a template, if any and if this template defines a type
- Specific type : either request or incident,
- Equals to the answer to the question : the value is set from a request type question.

### Associated elements

Sets associated elements to the generated ticket

None : self explanatory, Specific asset : an asset set by the designer of the form, Equals to the answer to a question : an asset set by the requester, from a question Last valid answer : an asset set by the requester, from the last visible question eligible for this purpose

### Category

Sets the category of the generated ticket

- Category from template or none : sets the category from a template if any and if this template has a predefined category. None in all other cases,



- Specific category : a category set by the form's designer,
- Equals to the answer to a question : a category choosen by the requester,
- lastr valid answer : a category from the last visible question of type category,

### **Urgency**

- Urgency fro template or none : sets the urgency from a template if any and if this template has a predefined urgency. None in all other cases,
- Specific urgency : an urgency set by the form's designer
- Equals to the answer to a question : a category choosen by the requester,

### **Location**

- Location from template or none : sets the location from a template if any and if this template has a predefined location. None in all other cases,
- Specific location : a location set by the form's designer
- Equals to the answer to a question : a category choosen by the requester,

### **Condition to show the target**

The form's designer may restrict creation of the ticket depending on conditions.

## **14.5.2 Target change**

A target change generates a change

K<< change >>K

Target change

Edit a destination
Name
change

Target change
Change title \*
change
Description \*
##FULLFORM##
Impacts
Control list
Deployment plan
Backup plan
Checklist
Destination entity
Current active entity
Time to resolve
Time to resolve
Ticket category
None
Urgency
Medium
Cancel Save

Actors
Requester +
Watcher +
Assigned to +
Form requester
Form validator

List of available tags

Question	Title	Answer	Section
Full form	-	##FULLFORM##	-

The change is built from scratch. You may

- customize the title, description, impact, control list, deployment path, backup plan and checklist fields using questions and answers,
- set the urgency from a value or a question
- choose the entity of the change among many policies
- set the time to resolve
- define the actors of the change
- set the category of the change
- define conditions to meet to generate the target ticket



## 14.6 Exporting and importing forms

### 14.6.1 Abstract

Formcreator allows exporting and importing forms between instances of GLPI.

This feature is designed to let administrators develop forms on a testing environment and copy them on a production environment.

When a form is being imported the plugin searches for a matching form in the database. This is done with the help of a random identifier created when a new form is instantiated. Each sub object (sections, questions, conditions, targets)

also have a random identifier. This helps the plugin to find a matching item and update it if necessary. Thanks to this mechanism Formcreator is able to update a form when it is imported more than once.

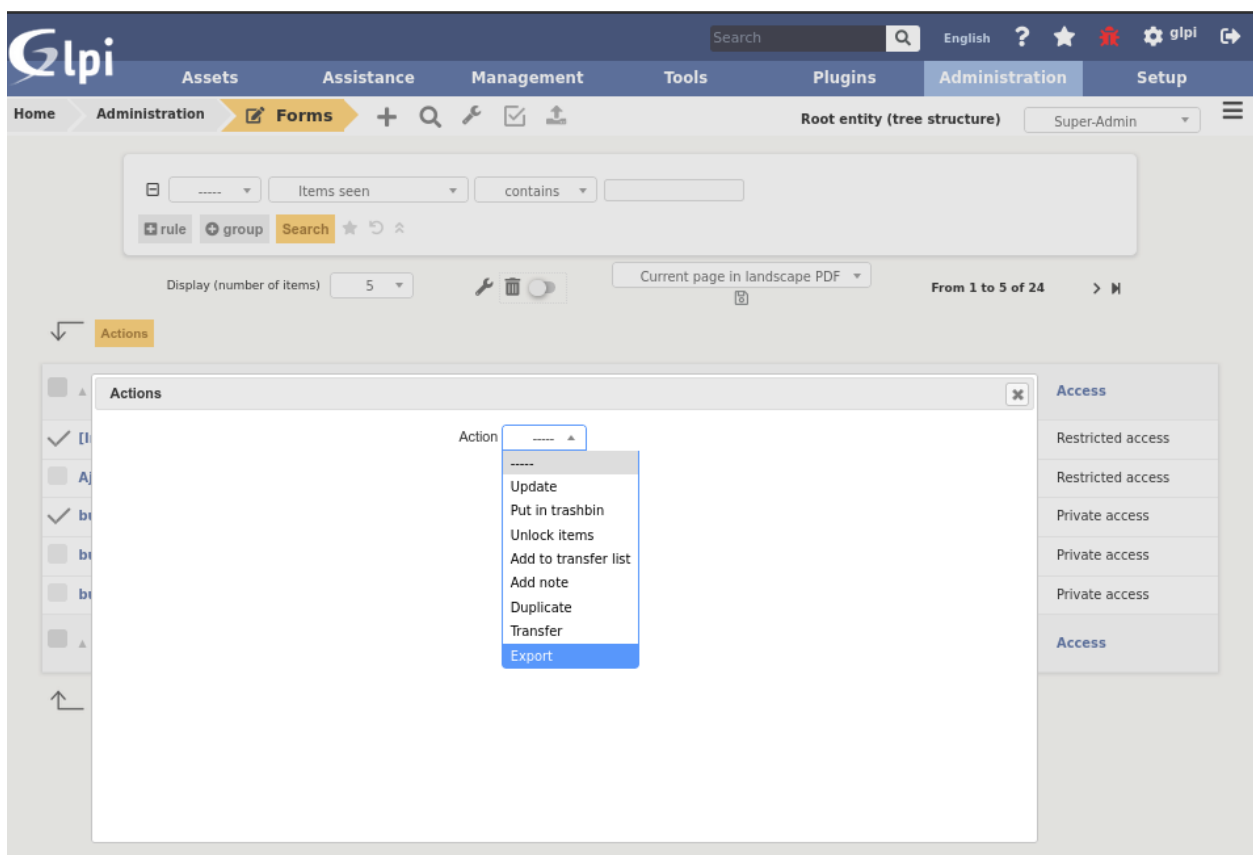
## 14.6.2 Limitations

A form may contain relations to objects of GLPI itself (mostly users, groups, categories, entities). There is currently no reliable way to maintain relations between forms and GLPI's objects or assets accross instances. It is therefore highly recommended to develop forms on a testing environemnt with a reasonably recent copy of the production database.

The export / import feature does not support importing forms accros different versions of Formcreator. The version 2.10 shows a warning if it detects a attempt of importing forms from a previous version. In version 2.11 such imports are not allowed.

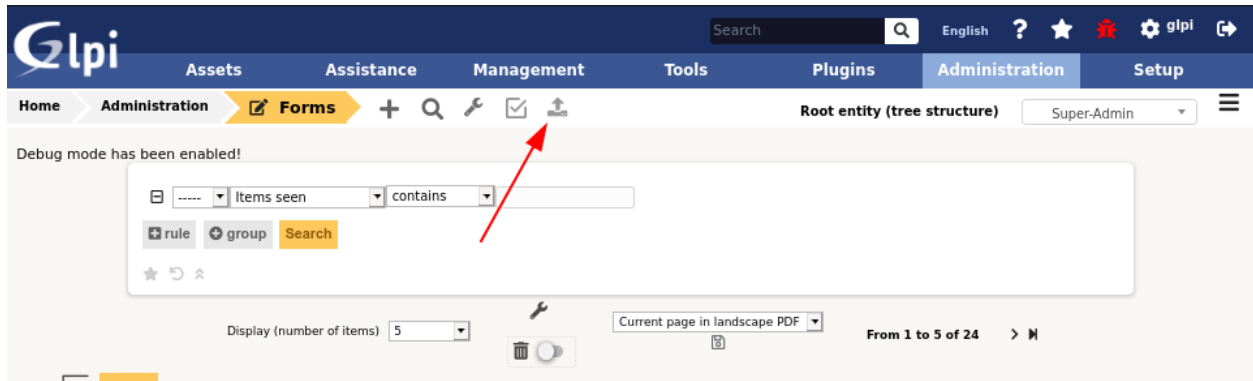
## 14.6.3 Exporting forms

1. Navigate to **Administration > Forms**
2. Use massive actions to export one or more forms. Clicking on the **Post** button will start the download of a JSON formatted file. This file contains all forms selected for the export.

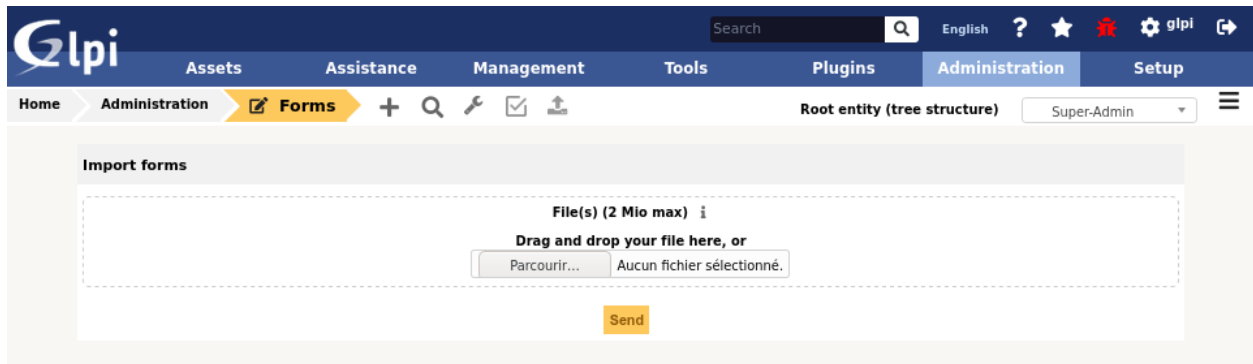


## 14.6.4 Importing forms

1. Navigate to **Administration > Forms**
2. Use the toolbar to import forms.



3. Select one or more JSON files created with the same version of Formcreator and validate.



4. Depending on the complexity and the quantity of forms the process may be slow. Please be patient.

5. Carefully check the messages when the import completes. You may get warnings and errors.

## 14.6.5 Import behavior

### Entities

When Formcreator imports a form, it checks that the entity of the form exists and the user has right to update entities. If not, the form is ignored and trigger a warning.

### Form categories

Form categories are created if they don't exist yet.

### Users and groups

The users and groups found in target actors must exist. If not, the import of the form will stop with an error, leading to an incomplete form. Other forms will be imported.



## 14.7 Contributing

### 14.7.1 Adding new features

#### You are a developer ?

You can help us adding new features by forking the GitHub repository.

Develop the new feature on your branch and then, ask for a [pull request](#) to merge your development into the repository.

You can also help us debug the [declared issues and requested features](#).

#### You are not a developer ?

You can submit your ideas of new feature by [creating an issue](#) or adding comment on an existing issue (to add explanations or just to say « I'm also interested »).

See also the [roadmap](#)

### 14.7.2 Help us debug

If you are a developer and want to code the fix yourself, just read the paragraph bellow...

But if you are not a developer or don't want to develop fixes yourself, you can still help us by creating [issues](#). Indicate your GLPI and plugin's version and steps to reproduce for a faster and easier fix.

### 14.7.3 Documentation

You can fork the [documentation repository](#) to add new contents or fix some issues in existing content.

Make your changes on your branch and then, ask for a [pull request](#) to merge it into the repository.

### 14.7.4 Translations

If you want Formcreator to be available in your native language and have a little time, you can help us :

Join us on [Transifex](#).

**Current available languages :** Czech (Czech Republic), English (United Kingdom) , French (France), German (Germany), Hungarian (Hungary), Polish (Poland), Portuguese (Brazil), Romanian (Romania), Russian (Russia), Spanish (Argentina), Turkish (Turkey)



## 14.8 Team

### 14.8.1 Developers

- Jérémy MOREAU,
- Alexandre Delaunay,
- François Legastelois,
- Thierry Bugier

## 14.8.2 Testers

- Manu1400,
- dgsn,
- Max6Devil,
- Kahalisto,
- satyan01,
- biet-j,
- reg25,
- Gilmor49,
- nblaisonneau,
- CHCI74,
- salgueirosa,
- jotafe,
- Ruiseart,
- babalou,
- blienard,
- proprux,
- ThedarksideoftheForce,
- AdAugustaPerAngusta,
- MaxG89,
- klodnitsky,
- consolko,
- Boris31,
- fire2418,
- osfrance,
- kaioisdead,
- KevinSupertramp,
- matth974,
- J-n-s,
- wawax,
- Sismic,
- nicholaseduardo
- and much more...

## 14.8.3 Translators

### French (France)

- Jérémy MOREAU,
- Alexandre DELAUNAY,
- Walid Nouh,
- Thierry Bugier Pineau,
- Benoit Le Rohellec,
- Emmanuel Haguet

### Turkish (Turkey)

- Kaya Zeren

### English (United Kingdom)

- Jérémy MOREAU,
- Andrejs Klodnickis

### Polish (Poland)

- Sismic,
- Ryszard Jeziorski,
- awiamo,
- Grzegorz Kaniewski

### Russian (Russia)

- Alexey Petukhov,
- Igor Nikitin,
- Andrejs Klodnickis

### Portuguese (Brazil)

- Pedro de Oliveira Lira,
- Rafael Viana

### Hungarian (Hungary)

- Laszlo Czirbesz]

### Spanish (Argentina)

- Luis A. Uriarte

### Romanian (Romania)

- Doru DEACONU

### Czech (Czech Republic)

- David Stepan







- Sources link : <https://github.com/pluginsGLPI/datainjection>
- Download : <https://github.com/pluginsGLPI/datainjection/releases>

### 15.1 Requirements for latest version

This plugin requires :

- PHP 7.2 or higher
- GLPI >= 9.4

### 15.2 Features

This plugin allows data import into GLPI using CSV files.

It allows to create models of injection for a future re-use. It's been created in order to :

- Import data coming from others asset management softwares
- Inject electronic delivery forms

Data to be imported using the plugins are :

- Inventory data (except softwares and licenses),
- Management data (contract, contact, supplier),
- Configuration data (user, group, entity).

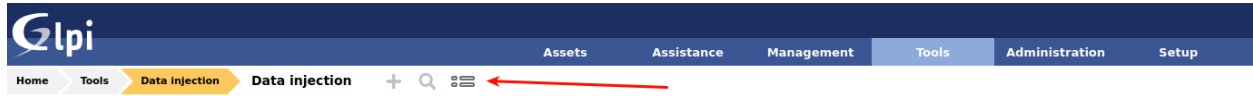
### 15.3 Install the Plugin

- Uncompress the archive.
- Move the `escalade` directory to the `<GLPI_ROOT>/plugins` directory
- Navigate to the *Configuration > Plugins* page,
- Install and activate the plugin.

## 15.4 Configuration

You will access the datainjection configuration from the *Tool > File injection*.

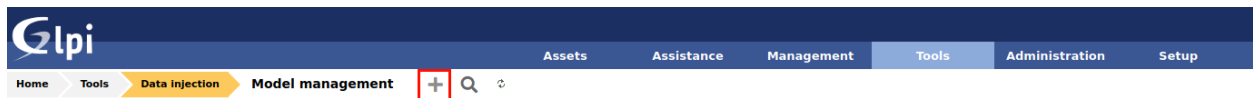
Clic here to manage model



## 15.5 Create new model

Fist step you need to create model, for this example we import Computer

Clic here to create new model



And fill form

- Name : define a model Name
- Visibility : is private or not for other user
- Entity / sub entity : model visibility for entity
- Comments : model comment
- Type of data : type of data to import
- Allow lines creation : yes or not
- Allow lines update : yes or not
- Allow creation of dropdowns : if dropdown value not exist, let's create It
- Dates format : date format in CSV file
- Allow update of existing fields : yes or not
- Float format : float format in CSV file
- Try to establish network connection is possible : yes or not
- Port unicity criteria : define unicity field fir port

After model creation it's possible to : \* Define if header is present \* Change file delimiter : default -> « ; »

## 15.6 Inject your CSV file

Send to GLPI your CSV file with computer data

content of CSV file for this documentation

```
Name;Type;Model;Manufacturer;serial
Desktop-ARTY;Desktop;Dell Inspiron;Samsung;567DFG45DFG
Laptop-QUER;Laptop;Dell XPS;Samsung;345UKB78DGH
```

## 15.7 Mapping CSV column and object field

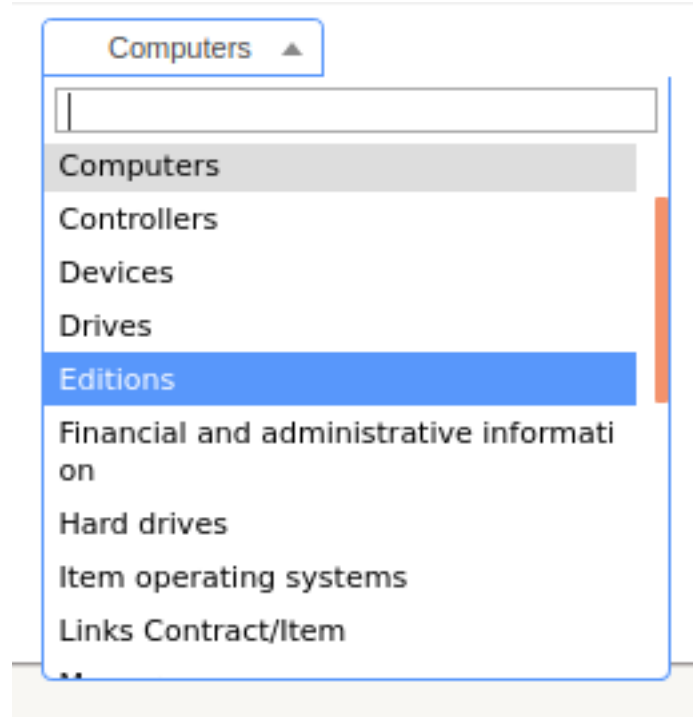
For each column of your CSV file you must select the table and the corresponding field in GLPI

**Note :** You need to define link field. The plugin will search on this link field to know if object need to be added or updated

Header of the file	Tables	Fields	Link field
Name	Computers	Name	<input checked="" type="checkbox"/>
Comment	Computers	Comments	<input type="checkbox"/>
Type	Computers	Type	<input type="checkbox"/>
Model	Computers	Model	<input type="checkbox"/>
Manufacturer	Computers	Manufacturer	<input type="checkbox"/>
serial	Computers	Serial number	<input checked="" type="checkbox"/>

The dropdown list contains other tables, which allows it to import, for example, the financial and administrative informations during computer import.

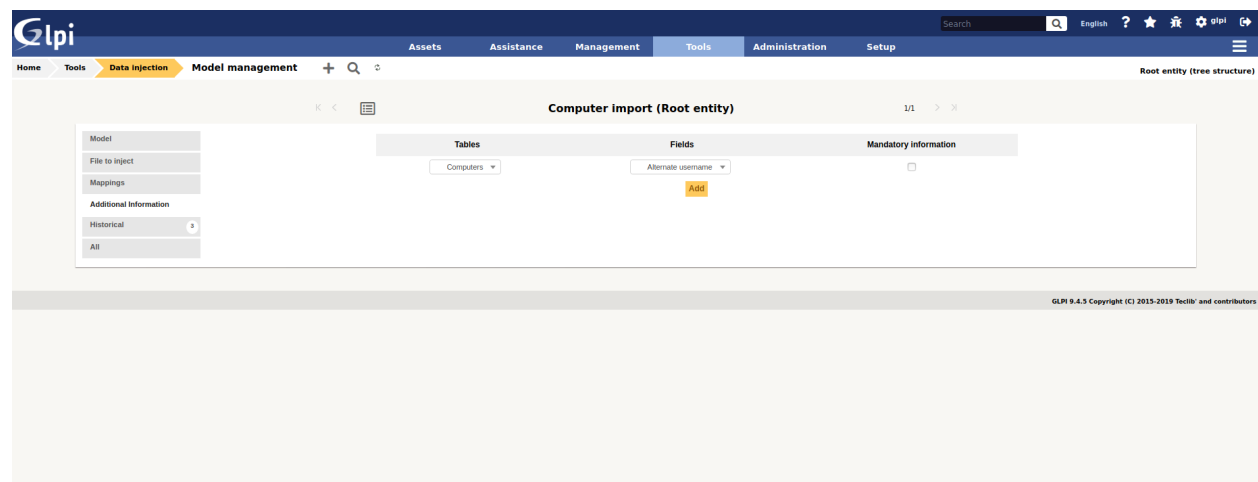
Each type of data (Computer, Monitor, User) have different options to import other data



## 15.8 Additional data

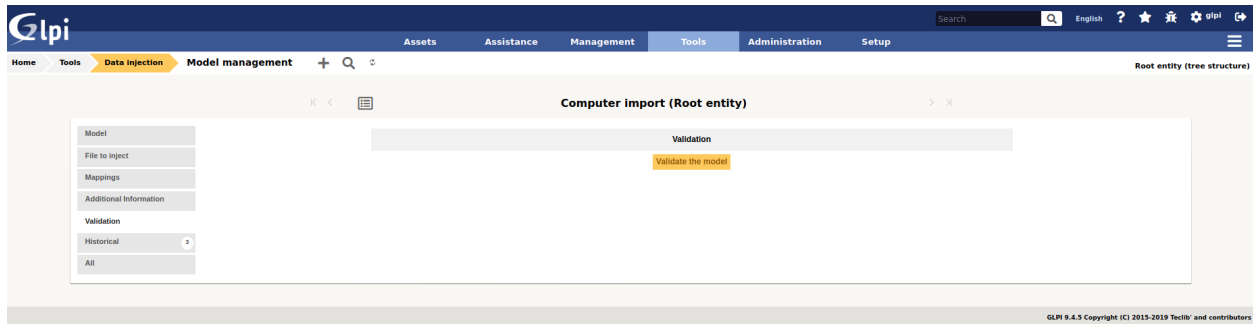
You can define additional data to be imported, it will be requested during import.

Each can be flag as mandatory



## 15.9 Validate model

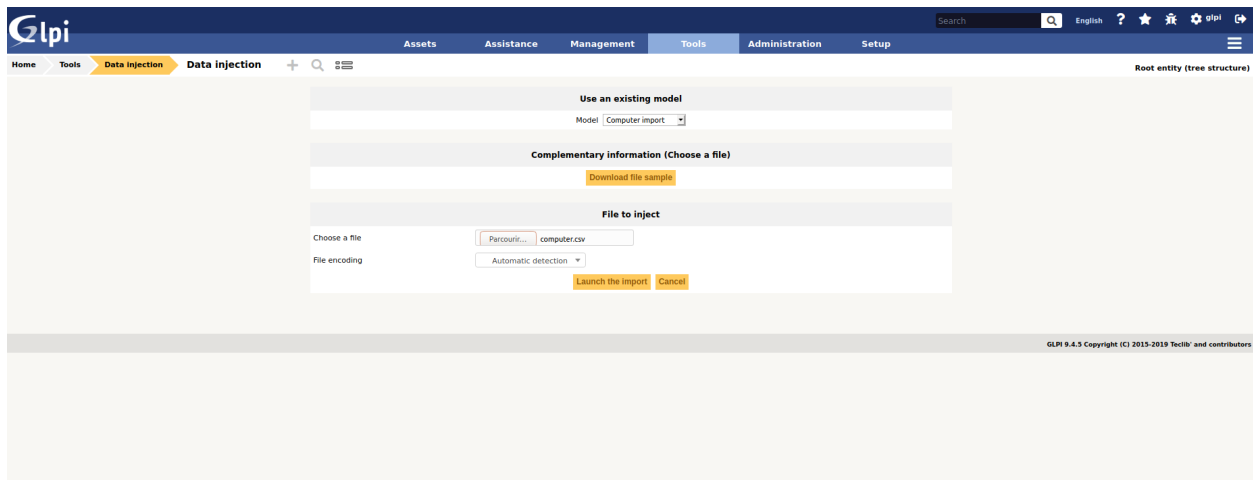
After configuration you can validate model



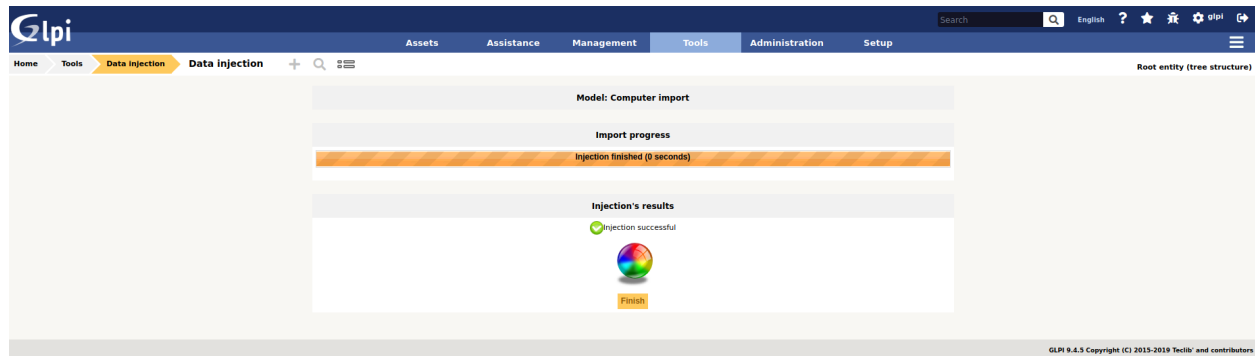
## 15.10 Execute import

You will access the model from the *Tool > File injection*.

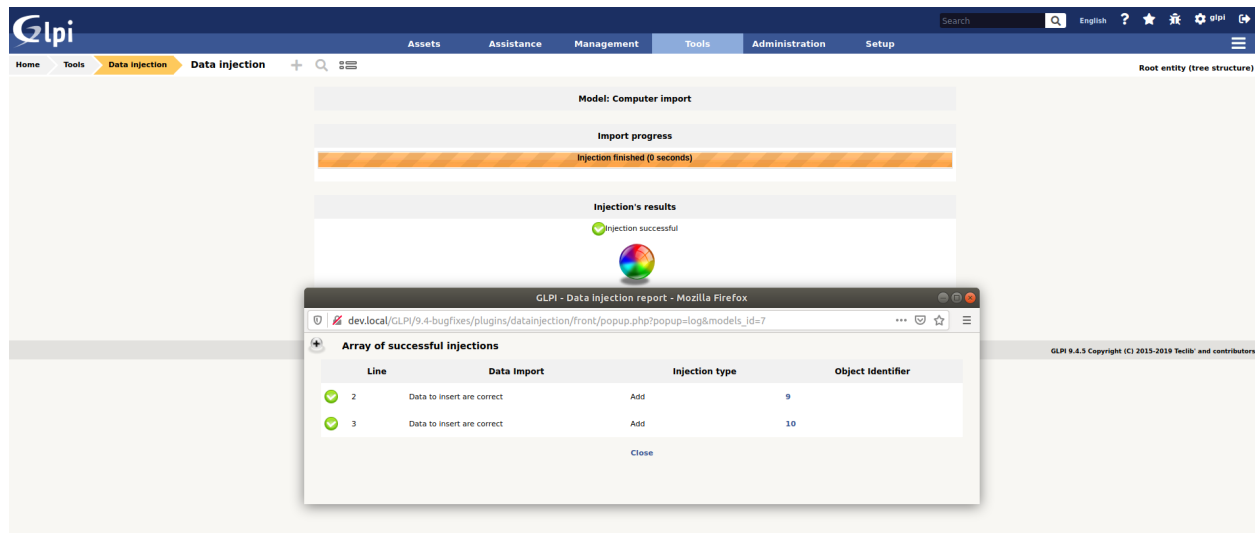
Select a model, select CSV file, and run import



## 15.11 Import result



After import process, plugin show you result, clic on spere to get more details



- Code source : <https://github.com/pluginsGLPI/oauthimap>
- Téléchargement : <https://github.com/pluginsGLPI/oauthimap/releases>

### 16.1 Fonctionnalités

Ce plugin permet la connexion Oauth pour les collecteurs mail.

Services de mail supportés :

- Google (G Suite and Gmail),
- Microsoft (Office 365 via Azure AD).

### 16.2 Installer le plugin

Le plugin est disponible au sein du marketplace de GLPI.

Vous pouvez aussi l'installer manuellement :

- Décompressez l'archive.
- Déplacez le dossier `oauthimap` dans le dossier `<GLPI_ROOT>/plugins`.
- Naviguez vers la page *Configuration > Plugins*.
- Installez et activez le plugin.

### 16.3 Configurer votre Application Oauth IMAP

Pour pouvoir établir une connexion à une boîte mail en utilisant l'authentification Oauth, vous devez au préalable configurer votre *Application Oauth IMAP* :

- Enregistrez votre application externe sur la console de gestion du fournisseur de service ;
- Créez un élément dans *Configuration > Applications Oauth IMAP* en utilisant les champs *id* et *secret* de l'application du fournisseur de service ;

- Copiez l'URL de retour depuis la page de configuration de l'Application Oauth IMAP et saisissez là dans la console du fournisseur de service (GLPI sera appelé à la fin du processus d'authentification).

**New item - Oauth application**

Name	<input type="text" value="Azure IMAP"/>
Active	<input type="button" value="Yes"/>
Oauth provider	<input type="button" value="Azure"/> ?
Client ID	<input type="text" value="86182394-075c-4c4a-a89d"/>
Client secret	<input type="text" value="secret-key-of-my-oauth-application"/>
Tenant ID	<input type="text" value="b6c80bbe-aaba-41ea-9235-9c0ec373fbfe"/>
Callback url	<input type="text" value="https://glpi.intra/marketplace/oauthimap/front/authorization.callba"/>

## 16.4 Créer une autorisation

Une fois votre Application Oauth IMAP créée, vous devrez créer une autorisation qui sera utilisée lors de l'authentification à votre boîte mail. Pour se faire, allez dans l'onglet *Autorisation Oauth* de votre Application Oauth IMAP, et cliquez sur *Créer une autorisation*.

**Azure IMAP - ID 1** Actions ▼ 1/1

Oauth application	<input type="button" value="+ Create an authorization"/>	
Historical 13		
Receivers		
Oauth authorization 1		
All		

Email	Expires at	
user@yourdomain.onmicrosoft.com	2020-10-09 11:50	<input type="button" value="Delete"/>

Vous pouvez créer une autorisation pour chacune des boîtes mail que vous souhaitez utiliser dans des collecteurs mail. Quand vous créez une autorisation pour une boîte mail, toute autorisation existante pour cette même boîte est défaussée.

## 16.5 Configurer votre collecteur mail

Pour s'authentifier via Oauth à votre collecteur mail, vous devez changer la configuration de celui-ci :



- choisissez votre Application OAuth IMAP dans la première liste déroulant des *Options de connexion*;
- puis choisissez une autorisation existant dans la liste déroulante *Identifiant* ou choisissez *Créer une autorisation pour un autre utilisateur*.

**New item - Receiver**

Name (Email address)

Active

Server

Connection options

Incoming mail folder (optional, often INBOX)

Port (optional)

Connection string

Login

Accepted mail archive folder (optional)

Refused mail archive folder (optional)

Maximum size of each file imported by the mails receiver

Use mail date, instead of collect one

Use Reply-To as requester (when available)

Add CC users as observer

Collect only unread mail

Comments

