

Les exercices se feront sur la machine marui.in2p3.fr . Vous y avez chacun un compte à votre nom.

## S'enregistrer dans une vo

La vo utilisée est la vo : vo.formation.idgrilles.fr qui est dédiée à la formation.

L'enregistrement se fait sur :

<https://cclcgvomsl01.in2p3.fr:8443/voms/vo.formation.idgrilles.fr/>

Avec votre certificat présent dans votre browser.

Cliquez sur le lien "New user registration" dans la liste à gauche.

- Remplissez le formulaire et cliquez sur le bouton "I have read and agree to the VO's Usage Rules".
- Vous allez recevoir un message électronique pour confirmer votre adresse électronique. (Vérifiez encore que le message n'est pas marqué comme SPAM .) Cliquez sur l'URL de confirmation indiquée.
- Le "VO Manager" doit valider votre demande. Une fois la demande validée, vous allez recevoir un message électronique.

## Transférer votre certificat

Pour l'exporter du navigateur:

Firefox : Sélectionnez le menu FireFox/Preferences . Cliquez sur "Avancer" à droite et l'onglet "Chiffrement". Cliquez sur le bouton "Afficher les résultats", sélectionnez votre certificat et utilisez la fonction "importer".

- Mozilla : Dans le menu Edit/Preferences allez dans l'onglet "Privacy \& Security" puis dans "certificates". Cliquez sur "Manage Certificates", et utilisez la fonction "backup".
- Netscape : Cliquez sur l'icône "sécurité" (ou passez par le menu Communicator/outils/Information sur la sécurité), puis cliquez sur le choix "vos certificats".
- Internet Explorer : Dans le menu Outils/Options Internet, allez dans l'onglet "contenu" et cliquez sur "certificats". Cliquez ensuite sur "exporter". Sélectionner exportation de la clé privée.

Il vous sera demandé un mot de passe que vous choisirez.

Créer un répertoire .globus : `mkdir .globus`

Y copier le certificat avec le nom `usercred.p12`

Définir les droits d'accès en lecture seule uniquement pour vous:

```
chmod 400 usercred.p12
```

Exercices:

Repérez les différents champs du certificat (sujet, validité...) en lisant son contenu avec la commande `openssl`:

```
openssl pkcs12 -in ~/.globus/usercred.p12 -nokeys -clcerts | openssl x509 -text
```

1. Quel est votre "Subject"?
2. Quelle est la date de fin de validité de votre certificat?

## S'identifiez sur la grille

### Creation d'un Proxy

Pour permettre à vos jobs d'accéder aux services de la grille, vous devez avoir un proxy. Un proxy est un "clone" de votre certificat contenant des informations supplémentaires, telles que la VO à laquelle vous appartenez (ces extensions sont souvent appelées extensions VOMS). Les proxies sont valables pour une durée limitée, normalement 12 à 24 heures.

Les principales commandes permettant d'obtenir un proxy et de le gérer sont :

- voms-proxy-init - -voms VO : permet de créer un proxy à partir du certificat. Cette commande demande le mot de passe protégeant la clé privée créée à l'étape précédente. **Il est très important d'utiliser l'option - -voms. VO est le nom de votre VO.**

*Note: le message Cannot find file or dir: ... /.glite/vomses est normal.*

- voms-proxy-info : affiche les informations sur le proxy existant. Avec l'option -all, permet d'afficher la liste des groupes et rôles activés.
- voms-proxy-destroy : supprime un proxy encore valide.

Il existe aussi des commandes myproxy-\*. Elles permettent le renouvellement automatique des proxies pour des jobs de très longue durée. Leur utilisation est présentée dans la partie de ce tutoriel sur [la soumission de jobs](#).

Exercices :

1. Demander un proxy à l'aide de la commande voms-proxy-init. Par exemple:  
`voms-proxy-init - -voms vo.formation.idgrilles.fr`
2. Affichez les informations du proxy créé. Les informations intéressantes sont le "subject", l'équivalent de votre "username" dans la grille, le "timeleft", le temps durant lequel le certificat est encore valable, et le nom de fichier pour le proxy.
3. Supprimez explicitement le proxy. Cette commande supprime le proxy dans l'ordinateur local. Les copies éventuellement envoyées avec vos jobs ne sont pas affectées. Vérifiez que le proxy a bien été supprimé avec la commande voms-proxy-info.
4. Regardez les autres options disponibles avec la commande voms-proxy-init. (Utilisez man ou l'option --help.) On peut changer, par exemple, la durée de vie et la taille (nombre de « bits ») du proxy. Normalement les valeurs par défauts sont correctes.

# Systeme d'information

## La Commande lcg-infosites

gLite fournit la commande lcg-infosites pour interroger le BDII. Elle cache la complexité de la syntaxe de la ldapsearch. Le but principal de la commande lcg-infosites est d'identifier les ressources disponibles pour une VO particulière. Pour cela on utilise l'option --vo voname. Le principal paramètre de cette commande est le type de ressource cherchée : ce, se, wms, lfc...

Utiliser l'option - -help pour obtenir la liste de tous les types de service.

*Note: l'utilisation de la commande lcg-infosites ne nécessite pas d'avoir un proxy.*

Exercices :

1. Identifier la liste de tous les *computing elements* (CE) accessibles à la VO vo.formation.idgrilles.fr.
2. Identifier la liste de tous les *storage elements* (SE) accessibles à la VO vo.formation.idgrilles.fr
3. Identifier la liste de tous les WMS (type wms) accessibles à la VO vo.formation.idgrilles.fr
4. Identifier la liste de tous les serveurs LFC accessibles à la VO vo.formation.idgrilles.fr

## Soumission de jobs

### Principales Commandes

Les commandes permettant de soumettre et gérer un job commencent toutes par glite-wms-job- et utilise les services d'un gLite WMS.

Les principales commandes sont :

- glite-wms-job-submit : soumission d'un job. Cette commande nécessite un *delegation proxy* qui peut être créé automatiquement si on utilise l'option -a ou être créé avec la commande glite-wms-job-delegation-proxy et être spécifié avec l'option -d identifieur. L'utilisation de -d doit être préférée si on soumet plusieurs jobs car elle est beaucoup plus efficace.
- glite-wms-job-status : affichage de l'état d'un job (en attente, en cours d'exécution, terminé...). L'option --all permet d'afficher l'état de tous les jobs de l'utilisateur, si le LB (*Logging and Bookkeeping*, l'un des services du WMS) utilisé fait partie des LBs configurés par défaut sur l'UI.
- glite-wms-job-output : récupération de l'output du job. Le répertoire utilisé pour stocker les résultats dépend de la configuration du site. Au LAL, le répertoire est ~/JobOutput/. Ce répertoire **doit exister** avant l'exécution de la commande : il faut le créer si nécessaire.
- glite-wms-job-logging-info : affichage d'informations détaillées sur l'exécution du job et les éventuelles erreurs rencontrées.
- glite-wms-job-delegate-proxy -d identifieur : obtention d'un delegation proxy pour soumettre plusieurs jobs. identifieur est un nom arbitraire qui sera utilisé pour désigner le *delegation proxy* et l'utiliser dans avec les commandes comme glite-wms-job-submit.

*Note: l'utilisation des commandes glite-wms-job-xxx nécessite d'avoir un proxy valide.*

Toutes les commandes ont une aide en ligne accessible en utilisant l'option --help ou la commande man.

La commande glite-wms-job-submit retourne un *jobid* qui est un URL. Ce *jobid* devra être utilisé pour désigner le job dans les autres commandes. Il doit être conservé sans quoi il n'est plus possible d'interagir avec le job (connaître son état ou récupérer son output). Quand on soumet plusieurs jobs, il peut être plus pratique d'utiliser l'option -o fichier lors du submit : dans ce cas, le *jobid* sera écrit dans le fichier. Le même fichier pourra être utilisé à la place du paramètre *jobid* dans les autres commandes en utilisant l'option -i.

Les commandes glite-wms-job-xxx nécessitant d'avoir un proxy valide, il n'y a pas besoin de spécifier la VO utilisée pour ces commandes. Elle est obtenue à partir du proxy.

Il existe également des APIs ([Java](#), C, C++) qui permettent la gestion des jobs depuis une application. Ce tutorial ne couvre pas la description de ces APIs.

## **Soumission d'un Job "Hello World" \_**

Cette exercice consiste à soumettre un job très simple qui écrira "Hello World" dans le fichier d'output. Le job en lui-même n'a pas d'intérêt particulier mais permet d'illustrer la soumission et le suivi de l'exécution d'un job. Un job grille est décrit dans un langage particulier appelé [JDL](#), décrit dans la partie suivante.

1. Si ce n'est pas déjà fait, créez un proxy à l'aide de la fonction voms-proxy-init.
2. Soumettez le job HelloWorld.jdl se trouvant dans le matériel du tutorial récupéré précédemment, en utilisant la commande glite-wms-job-submit. Sauver le *jobid* retourné ou utiliser l'option -o.
3. Vérifiez le statut du job en utilisant la commande glite-wms-job-status. On peut utiliser la commande watch pour exécuter une commande dans une boucle (pour effectuer la commande glite-wms-job-status toutes les 15 secondes, utilisez watch -n 15 glite-wms-job-status <jobid> et taper ctrl-c pour sortir).
  - Suivez les modifications de l'état du job jusqu'à l'état "Done(Success)".
  - Combien d'états différents pouvez-vous distinguer ?
  - Si le job se termine dans l'état "Aborted", c'est qu'il y a eu une erreur. On peut trouver plus d'informations avec la commande glite-wms-job-logging-info.
4. Lorsque le job est terminé (Status: Done (Success)), récupérez les données générées à l'aide de la commande glite-wms-job-output.
5. Vérifiez que tout s'est déroulé correctement en consultant les fichiers std.out et std.err. Le fichier std.err doit être vide et std.out doit contenir "Hello World". Cette procédure simple est utilisée pour le suivi de tous les jobs.

# Gestion des données

Les services de grille impliqués dans la gestion des données sont le LFC (file catalogue) et les SE storage element.

LFC: Ce service gère les noms des fichiers donnés par les utilisateurs et leur réplicas. Les noms de fichier sont comme des fichiers Unix. Un fichier peut se trouver à plusieurs endroits (réplica).

SE: C'est le service qui gère les données. Il y a différentes implémentations de ce service qui peuvent s'interfacer avec différents médias (disques, bandes etc..)

## Utilisez le file catalog

Le service « File catalog » (aussi appelé replica catalogue) est implémenté dans glite en utilisant LFC. Le service LFC a des clients implémentés via des commandes lfc- Ces commandes sont disponibles sur les ui et les worker nodes.

En général les commandes lcg-utils sont plutôt utilisées. Une exception est la commande lfc-mkdir qui doit être utilisée pour créer des répertoires avant d'y créer des LFN.

```
export LFC_HOST=lfc-egge.in2p3.fr
```

Les commandes LFC principales sont:

- lfc-mkdir [-]: crée un répertoire dans LFC
- lfc-ls [-l]: similaire à la commande Unix ls. Donne le contenu du répertoire et peut donner d'autres informations.
- lfc-getacl / lfc-setacl: LFC namespace supports Posix-like ACLs. See the man page for more information on the syntax, in particular for lfc-setacl. A file name has one ACL but a directory has 2 different ACLs: the ACL controlling access to the directory and the default ACL applied to new files (new sub-directories inherit the parent ACL).

## Commandes principales:

Un ensemble de commandes sous le nom de lcg\_utils sont disponibles pour l'utilisateur, elles sont indépendantes de l'implémentation (donc valable pour tous les SE).

La plupart des commandes peuvent utiliser pour un fichier différents type d'identifiant.

Logical File Name (LFN): nom dans le catalogue de fichier , syntaxe Unix like et commençant par /grid/voname. A LFN doit être préfixé lfn:.

Dans notre cas /grid/vo.formation.idgrilles.fr/cppm2001

- GUID: identifiant unique associé à un fichier (ne change pas si LFN est changé ou renommé). A GUID doit être préfixé guid:.
- SURL: nom du replica sur un SE particulier . Il a un format URL like srm://se.host.name/path/to/file.

Main commands are:

- lcg-cr (copy and register): this command copies a file available locally (it can also copy a file from another SE) and create a new file on a SE, registering it in a file catalog. Copy destination **must be** a storage element. Even though the source is a file on a SE, this creates

a new file and not a new replica of the existing file. By default logical file name registered in the LFC is generated but it can be set explicitly using option `-l` (*directories* in the LFN must be created with `lfc-mkdir`, as in Unix they must be created with `mkdir` before being used). The storage element to use for the destination can be specified with option `-d`. If omitted, the default SE (as specified by environment variable `VO_VONAME_DEFAULT_SE`) for the VO is used. If `-d` is present but specifies only a host name, the file name on the SE is generated. It is recommended to use option `-v`` (verbose) to get details about the copy operation and ease the troubleshooting in case of errors.

- `lcg-cp` (copy): feature and syntax very similar to Unix `cp` command. Source and destination can be either a local file or a file on a SE, with any combination. Conversely to `lcg-cr` destination is not registered into a catalog and cannot be a LFN (the input file may be identified with a LFN or GUID if it resides on a SE). This command is mainly used to get a local copy of a SE file on a UI or WN.
- `lcg-rep`: this command allow to add a new replica to an existing file registered in a file catalog. Syntax is similar to `lcg-cr` but the source **must be** a LFN. But conversely to `lcg-cr`, this command updates an existing entry in the file catalog but doesn't create a new one.
- `lcg-lr` (list replica): list all the replicas associated with a given LFN or GUID.
- `lcg-ls` : feature and syntax very similar to Unix `ls` command.
- `lcg-del` : removes a file replica from a SE and optionally remove entry from LFC after removing all replicas.
- `lcg-lg` : returns the GUID associated with a LFN.

Toutes ces commandes reposent sur le BDII pour utiliser les défauts selon les vo. Si la BDII n'est pas disponible on peut utiliser l'options `--nobdii` mais il faut alors préciser beaucoup de paramètres.

Exercices :

1. Affichez les SE disponibles pour la VO `vo.formation.idgrilles.fr`, using command `lcg-infosites`
  - Combien de SEs possibles pour `vo.formation.idgrilles.fr` ?
  - Trouvez la même chose pour la VO `dteam`. Combien de SEs sont disponibles pour cette vo ?
2. Créez un répertoire à votre nom avec `lfc-mkdir`
3. Créez un fichier texte et le copier sur un SE en utilisant la commande `lcg-cr`, en utilisant option `-l` pour définir le nom logique . Essayez de définir le SURL (replica name on SE) aussi
4. Vérifiez que le fichier est présent dans le catalogue et lister ses réplica avec la commande `lcg-lr`. Cela doit vous donner le SURL pour tous les replicas (a SURL starts with prefix `srn:`).
5. Affichez toutes les informations de ce fichier avec la commande `lcg-ls`.
6. Trouvez le GUID associé aux fichiers précédents.
7. Répliquez le fichier sur un autre SE ouvert à notre vo (using the information got with `lcg-infosites`).
8. Vérifiez que le nouveau replica a été ajouté au catalogue avec la commande `lcg-lr` comme précédemment. Comparez les résultats.
9. Recopiez localement le fichier à partir du SE en utilisant la commande `lcg-cp`. Faites le en utilisant le GUID, le LFN et un replica SURL.
10. Supprimez un replica avec la commande `lcg-del` et vérifiez le résultats avec `lcg-lr`.
11. Supprimez l'autre replica avec la commande `lcg-del` et vérifiez le résultats avec `lcg-lr`.
12. Recréez un fichier avec 2 replicas et essayez de les supprimer en une fois. Essayez `lcg-lr` et `lcg-ls`.

**Utilisez le file catalog**

Le service « File catalog » (aussi appelé replica catalogue) est implémenté dans glite en utilisant LFC. Le service LFC a des clients implémentés via des commandes lfc- Ces commandes sont disponibles sur les ui et les worker nodes.

En général les commandes lcg-utils sont plutôt utilisées. Une exception est la commande lfc-mkdir qui doit être utilisées pour créer des répertoires avant d'y créer des LFN.

```
export LFC_HOST=lfc-egge.in2p3.fr
```

Les commandes LFC principales sont:

- lfc-mkdir [-]: crée un répertoire dans LFC
- lfc-ls [-l]: similaire à la commande Unix ls. Donne le contenu du répertoire et peut donner d'autres informations.
- lfc-getacl / lfc-setacl: LFC namespace supports Posix-like ACLs. See the man page for more information on the syntax, in particular for lfc-setacl. A file name has one ACL but a directory has 2 different ACLs: the ACL controlling access to the directory and the default ACL applied to new files (new sub-directories inherit the parent ACL).