

Table des matières

Installer un serveur DAoC sous Linux 3

Installer un serveur DAoC sous Linux

Les infos contenues dans cet article ont été testées avec Ubuntu 20.04. Le moteur de serveur de jeu utilisé est développé en C#. Il faut donc installer un environnement mono suffisamment récent et à jour pour le faire fonctionner. On va voir ici les étapes pour faire fonctionner tout ce bazar. On aura besoin des composants ci-dessous :

- base de données : MariaDB
- framework .NET/C# : [mono](#)
- dotnet pour linux : <https://dotnet.microsoft.com/download/dotnet/5.0>
- moteur de jeu : Dawn of Light : [DOLSharp](#)
- base de données du jeu (mobs, sorts, etc) : Eve Of Darkness

Packages à installer

```
wget https://packages.microsoft.com/config/ubuntu/21.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
dpkg -i packages-microsoft-prod.deb
apt-get update
apt-get install mono-complete mono-devel git unzip mariadb-server nuget dotnet-sdk-5.0
```

Création d'un utilisateur dédié :

On crée un user dol dont le home directory /home/dol servira de référence pour toutes les commandes passées.

Création de la base de données :

Après avoir installé MariaDB on peut créer la base de données (UTF8, important), l'utilisateur et on ajuste les droits.

```
CREATE DATABASE dol CHARACTER SET utf8;
CREATE USER 'dol'@'%' IDENTIFIED BY 'YourPassword';
GRANT SELECT , INSERT , UPDATE , CREATE , INDEX , ALTER , CREATE TEMPORARY TABLES , CREATE VIEW , SHOW VIEW , CREATE ROUTINE , ALTER ROUTINE , EXECUTE , DROP , DELETE \
ON `dol` . * TO 'dol'@'%' IDENTIFIED BY 'YourPassword';
```

Import des données :

Plusieurs bases de données sont disponibles, certaines plus ou moins à jour ou fidèles par rapport aux serveurs officiels. Ici on part sur la base Eve of Darkness qui est un énième fork mais qui est plutôt fidèle à ce qu'on trouve sur <https://camelot.allakhazam.com/>. Je détaille comment récupérer et importer les données, c'est pas forcément très intuitif :

```
git clone https://github.com/Eve-of-Darkness/db-public.git
cd db-public
mv src/scripts/config/config.example.yml src/scripts/config/config.yml
cd src/scripts/bin/linux/
./concat # ou ./concat --update-only en cas de mise à jour
mysql -u dol -p dol < ./public-db.sql
```

Récupération et compilation du moteur de jeu

Encore une fois ce n'est pas forcément très bien documenté et on ne cross-compile pas du C# sous Linux tous les jours. On récupère DOLsharp. On utilisera le client [1.127c](#).

Compilation

```
mkdir /home/dol/work
cd /home/dol/work
git clone https://github.com/Dawn-of-Light/DOLSharp.git
cd DOLSharp
nuget restore -SolutionDirectory ./

for i in DOLBase DOLDatabase GameServer DOLServer Tests GameServerScripts DOLConfig
do
mkdir -p /home/dol/work/DOLSharp/build/${i}/Release
ln -sf /home/dol/work/DOLSharp/build/${i}/project.assets.json /home/dol/work/DOLSharp/build/${i}/Release/project.assets.json
done

MONO_IOMAP=case msbuild /p:Configuration=Release "Dawn of Light.sln"
ln -sf /home/dol/work/DOLSharp/Release /home/dol/live
```

On crée un lien symbolique vers le moteur compilé, c'est plus simple. Lors de la première compilation il est normal d'avoir des erreurs. Avant la première exécution du moteur il faut créer un répertoire config avec le fichier serverconfig.xml dont voici le contenu ci-dessous. Les paramètres importants sont :

- *EnableUPNP* qui doit être à False ;

- la chaîne `DBConnectionString` qui doit contenir les infos de connexion à la base de données. Le `DBType` est quant à lui positionné à `MySQL`.

```
dol@dolct:~$ cat /home/dol/live/config/serverconfig.xml

<?xml version="1.0" encoding="utf-8"?>
<root>
  <Server>
    <Port>10300</Port>
    <IP>0.0.0.0</IP>
    <RegionIP>0.0.0.0</RegionIP>
    <RegionPort>10400</RegionPort>
    <UdpIP>0.0.0.0</UdpIP>
    <UdpPort>10400</UdpPort>
    <EnableUPnP>False</EnableUPnP>
    <DetectRegionIP>True</DetectRegionIP>
    <ServerName>Dawn Of Light</ServerName>
    <ServerNameShort>DOLSERVER</ServerNameShort>
    <LogConfigFile>./config/logconfig.xml</LogConfigFile>
    <ScriptCompilationTarget>./lib/GameServerScripts.dll</ScriptCompilationTarget>
    <ScriptAssemblies>System.dll,System.Xml.dll</ScriptAssemblies>
    <EnableCompilation>True</EnableCompilation>
    <AutoAccountCreation>True</AutoAccountCreation>
    <GameType>Normal</GameType>
    <CheatLoggerName>cheats</CheatLoggerName>
    <GMActionLoggerName>gmactions</GMActionLoggerName>
    <InvalidNamesFile>./config/invalidnames.txt</InvalidNamesFile>
    <DBType>MySQL</DBType>
    <DBConnectionString>Server=localhost;Database=dol;User ID=dol;Password=YourPassword;Treat Tiny As Boolean=False</DBConnectionString>
    <DBAutosave>True</DBAutosave>
    <DBAutosaveInterval>5</DBAutosaveInterval>
    <CpuUse>1</CpuUse>
  </Server>
</root>
```

Customisation de la base de données

```
UPDATE `dol`.`ServerProperty` SET `Value`='True' WHERE `Key`='disable_tutorial';
UPDATE `dol`.`ServerProperty` SET `Value`='Bienvenue sur le serveur DAoC NDLP !' WHERE `Key`='motd';
UPDATE `dol`.`ServerProperty` SET `Value`='Bienvenue sur le serveur DAoC NDLP !' WHERE `Key`='starting_msg';
UPDATE `dol`.`ServerProperty` SET `Value`='1.5' WHERE `Key`='xp_rate';
UPDATE `dol`.`ServerProperty` SET `Value`='1' WHERE `Key`='starting_level';
UPDATE `dol`.`ServerProperty` SET `Value`='1.5' WHERE `Key`='money_drop';
UPDATE `dol`.`ServerProperty` SET `Value`='True' WHERE `Key`='allow_all_realms';
UPDATE `dol`.`ServerProperty` SET `Value`='True' WHERE `Key`='allow_all_realms_df';
UPDATE `dol`.`ServerProperty` SET `Value`='0.30' WHERE `Key`='health_regen_rate';
UPDATE `dol`.`ServerProperty` SET `Value`='0.75' WHERE `Key`='endurance_regen_rate';
UPDATE `dol`.`ServerProperty` SET `Value`='0.75' WHERE `Key`='mana_regen_rate';
UPDATE `dol`.`ServerProperty` SET `Value`='True' WHERE `Key`='allow_train_everywhere';
```

Lancement et contrôle du bon fonctionnement

```
cd /home/dol/live
mono --debug --gc=sgen --server ./DOLServer.exe
```

Il est 'normal' d'obtenir des erreurs et warnings. A la fin on doit obtenir :

```
[11:04:11,901] Server is now listening to incoming connections!
[11:04:11,906] Start Memory base.Start(): 1537MB
[11:04:11,906] base.Start(): True
[11:04:11,909] Finish Memory base.Start(): 1537MB
[11:04:12,670] GameServer is now open for connections!
> [11:04:21,765] -stats- Mem=1140MB Clients=0 Down=0kb/s (0MB) Up=0kb/s (0MB) In=0pck/s (0K) Out=0pck/s (0K) Pool=199/200(2) IOCP=200/200(2) GH/0H=39/1078 RegionTime1=14775t/s (1133) CPU=0.0%
```

Connexion au serveur

Pour se connecter au serveur créé, il est nécessaire de lancer le jeu via [Daoc Portal](#) après avoir récupéré le client DAoC officiel : https://files.ndlp.info/DAoC/DAoC_full.client.1.127c.zip

⇒ La version de la dll game.dll requise est : 1.127

Ressources utilisées dans cet article

<http://www.dolserver.net/>
<https://github.com/Dawn-of-Light/DOLSharp>
<http://losojos-001-site1.btempurl.com/Index.html> : site down , mirror dispo ici : <https://daoc.ndlp.info/losojos-001-site1.btempurl.com/Index.html>
<https://files.ndlp.info/DAoC/>

Un serveur DAoC est disponible à cette adresse : daoc-gs.ndlp.info:10300

From:
<https://unix-bck.ndlp.info/> - **Where there is a shell, there is a way**

Permanent link:
https://unix-bck.ndlp.info/doku.php/blog:installer_un_serveur_daoc_sous_linux?rev=1632213549

Last update: **2021/09/21 10:39**