

# Manuel d'installation et d'utilisation

## NTL-SysToolbox

**Version :** 1.0

**Date :** Février 2026

**Projet :** MSPR TPRES11 - Développement d'application NordTransit Logistics

### **Contexte académique**

*Ce document a été réalisé dans le cadre du projet **MSPR TPRES11***

*par le **Groupe 2** de la promotion **B3 ASRBD** de l'**EPSI Auxerre**.*

*Projet académique - Année 2025-2026*

## Table des matières

1. [Introduction](#)
2. [Prérequis système](#)
3. [Installation sur Linux](#)
4. [Installation sur Windows](#)
5. [Configuration initiale](#)
6. [Utilisation du Module 1 : Diagnostic](#)
7. [Utilisation du Module 2 : Sauvegarde WMS](#)
8. [Utilisation du Module 3 : Audit EOL](#)

## 1. Introduction

### 1.1 Présentation de l'outil

**NTL-SysToolbox** est un ensemble d'outils en ligne de commande développé pour automatiser les opérations d'administration système critiques de NordTransit Logistics.

**Fonctionnalités principales :**

- **Module 1** : Diagnostic des services critiques (AD/DNS, MySQL)
  - **Module 2** : Sauvegarde automatisée de la base WMS
  - **Module 3** : Audit d'obsolescence réseau avec détection EOL
- 

## 2. Prérequis système

### 2.1 Systèmes d'exploitation supportés

#### Linux :

- Debian 10, 11, 12
- Ubuntu 20.04 LTS, 22.04 LTS, 24.04 LTS
- CentOS 7, 8 Stream, Rocky Linux 9
- Red Hat Enterprise Linux 8, 9

#### Windows :

- Windows 10 (version 1809 ou supérieure)
- Windows 11
- Windows Server 2016, 2019, 2022

### 2.2 Prérequis logiciels

#### Obligatoires :

- Python 3.7 ou supérieur
- Accès Internet (pour installation des dépendances Python)

#### Module 3 uniquement :

- nmap (pour le scan réseau et détection OS)

#### Optionnels :

- Git (pour clonage depuis dépôt)

### 2.3 Droits et permissions

#### Linux :

- Utilisateur standard pour Modules 1 et 2
- **root ou sudo** pour Module 3 (nmap nécessite privilèges élevés)

#### Windows :

- Compte utilisateur standard pour Modules 1 et 2

- **Administrateur** pour Module 3
- 

## 3. Installation sur Linux

### 3.1 Vérification des prérequis

#### Étape 1 : Vérifier la version de Python

```
python3 --version
```

**Sortie attendue :**

```
Python 3.11.2
```

Si Python 3 n'est pas installé :

```
# Debian/Ubuntu
sudo apt update
sudo apt install python3 python3-pip -y
```

#### Étape 2 : Installer nmap (Module 3 uniquement)

```
# Debian/Ubuntu
sudo apt install nmap -y

Vérifier l'installation :
```bash
nmap --version
```

**Sortie attendue :**

```
Nmap version 7.93
```

### 3.2 Téléchargement de l'outil

Clonage depuis dépôt Git

```
cd ~  
git clone https://github.com/galaxie-offt/ntl-systoolbox.git  
cd ntl-systoolbox
```

## 3.3 Structure des fichiers

Après téléchargement, vérifier la présence des fichiers :

```
ls -l
```

**Fichiers attendus :**

```
-rw-r--r-- main.py  
-rw-r--r-- Module1.py  
-rw-r--r-- Module2.py  
-rw-r--r-- Module3.py
```

## 3.4 Rendre les scripts exécutables (optionnel)

```
chmod +x main.py Module1.py Module2.py Module3.py
```

## 3.5 Test de l'installation

```
python3 main.py
```

**Sortie attendue :**

```
NTL-SysToolbox - Menu principal  
1 - Module 1 : Diagnostic AD/DNS + MySQL + OS serveurs  
2 - Module 2 : Sauvegarde de la base WMS  
3 - Module 3 : Audit d'obsolescence réseau (EOL)  
0 - Quitter  
  
Votre choix :
```

☐ **Installation réussie** si le menu s'affiche correctement.

# 4. Installation sur Windows

## 4.1 Installation de Python

### Étape 1 : Télécharger Python

1. Aller sur <https://www.python.org/downloads/> (<https://www.python.org/downloads/>).
2. Télécharger **Python 3.11** ou version supérieure (Windows installer 64-bit)

### Étape 2 : Installer Python

1. Lancer l'installateur téléchargé
2. ☒ **Cocher "Add Python to PATH"** (Important !)
3. Cliquer sur "Install Now"
4. Attendre la fin de l'installation
5. Cliquer sur "Close"

### Étape 3 : Vérifier l'installation

Ouvrir **PowerShell** ou **Invite de commandes** :

```
python --version
```

Sortie attendue :

```
Python 3.11.8
```

## 4.2 Installation de nmap (Module 3 uniquement)

### Étape 1 : Télécharger nmap

1. Aller sur <https://nmap.org/download.html> (<https://nmap.org/download.html>).
2. Télécharger **nmap-7.95-setup.exe** (Latest stable release)

### Étape 2 : Installer nmap

1. Lancer l'installateur nmap
2. Accepter la licence
3. Garder les options par défaut
4. Installer **Npcap** quand demandé (requis pour Windows)
5. Terminer l'installation

### Étape 3 : Vérifier l'installation

```
nmap --version
```

**Sortie attendue :**

```
Nmap version 7.95
```

## 4.3 Téléchargement de l'outil

### Option 1 : Téléchargement direct

1. Créer un dossier C:\users\votreutilisateur\documents\ntl-systoolbox
2. Copier les fichiers Python depuis le git :

- main.py
- Module1.py
- Module2.py
- Module3.py

### Option 2 : Clonage Git (si Git installé)

```
cd C:\users\votreutilisateur\documents\  
git clone https://github.com/galaxie-offt/ntl-systoolbox.git  
cd ntl-systoolbox
```

## 4.4 Test de l'installation

Ouvrir **PowerShell** :

```
cd C:\users\votreutilisateur\documents\ntl-systoolbox  
python main.py
```

**Sortie attendue :**

```
NTL-SysToolbox - Menu principal  
1 - Module 1 : Diagnostic AD/DNS + MySQL + OS serveurs  
2 - Module 2 : Sauvegarde de la base WMS  
3 - Module 3 : Audit d'obsolescence réseau (EOL)  
0 - Quitter  
  
Votre choix :
```

□ **Installation réussie** si le menu s'affiche.

## 5. Configuration initiale

### 5.1 Configuration du Module 1 (Diagnostic)

Le Module 1 nécessite la configuration des paramètres de connexion AD/DNS et MySQL.

**Fichier à éditer :** `Module1.py`

**Paramètres AD/DNS :**

```
DOMAIN_CONTROLLERS = [  
    {"name": "DC01", "ip": "192.168.10.10", "os": "Windows Server 2019"},  
    {"name": "DC02", "ip": "192.168.10.11", "os": "Windows Server 2019"},  
]
```

**À adapter :**

- `name` : Nom du contrôleur de domaine
- `ip` : Adresse IP du DC
- `os` : Système d'exploitation (pour information)

**Paramètres MySQL :**

```
MYSQL_CONFIG = {  
    "host": "192.168.10.21",  
    "port": 3306,  
    "database": "wms"  
}
```

**À adapter :**

- `host` : Adresse IP du serveur MySQL
- `port` : Port MySQL (3306 par défaut)
- `database` : Nom de la base de données

### 5.2 Configuration du Module 2 (Sauvegarde WMS)

Le Module 2 demande les informations de connexion MySQL **de manière interactive** lors de son lancement.

**Aucune configuration préalable requise.**

Les paramètres suivants seront demandés au lancement :

- IP du serveur MariaDB/MySQL
- Nom de la base de données
- Nom d'utilisateur
- Mot de passe (saisie masquée)

## 5.3 Configuration du Module 3 (Audit EOL)

**Aucune configuration requise.**

Le Module 3 est autonome et demande les paramètres lors de l'exécution (plage IP à scanner, source de données EOL).

# 6. Utilisation du Module 1 : Diagnostic

## 6.1 Lancement du Module 1

**Linux :**

```
cd ~/ntl-systoolbox  
python3 main.py
```

**Windows :**

```
cd C:\ntl-systoolbox  
python main.py
```

**Menu principal :**

```
NTL-SysToolbox - Menu principal  
1 - Module 1 : Diagnostic AD/DNS + MySQL + OS serveurs  
2 - Module 2 : Sauvegarde de la base WMS  
3 - Module 3 : Audit d'obsolescence réseau (EOL)  
0 - Quitter  
  
Votre choix : 1
```

Saisir **1** puis **Entrée**.

## 6.2 Menu du Module 1



MODULE 1 - Diagnostic NTL

1 - Verifier l'etat des services AD/DNS

2 - Tester la base MySQL WMS-DB

3 - Afficher les informations d'OS des serveurs AD/DNS et MySQL

0 - Quitter

Votre choix :

## 6.3 Option 1 : Vérification AD/DNS

**Action** : Teste la disponibilité des ports critiques (LDAP, Kerberos, DNS) sur les contrôleurs de domaine.

**Saisir** : 1 puis **Entrée**

**Sortie attendue (exemple succès)** :

```
=== Test des services AD/DNS ===
```

```
DC01 (192.168.10.10):
```

```
Port LDAP (389): OK
```

```
Port Kerberos (88): OK
```

```
Port DNS (53): OK
```

```
DC02 (192.168.10.11):
```

```
Port LDAP (389): OK
```

```
Port Kerberos (88): OK
```

```
Port DNS (53): OK
```

```
Tous les services AD/DNS sont opérationnels.
```

```
Appuyez sur Entrée pour continuer...
```

**Sortie en cas d'erreur** :

```
DC01 (192.168.10.10):
```

```
Port LDAP (389): OK
```

```
Port Kerberos (88): KO (Timeout)
```

```
Port DNS (53): OK
```

```
⚠ Problème détecté sur DC01
```

**Interprétation** :

- **OK** : Le port répond, le service écoute

- **KO** : Pas de réponse (service arrêté, firewall, a réseau)

## 6.4 Option 2 : Test MySQL

**Action** : Teste la connexion complète à la base MySQL WMS avec authentification et requête test.

**Saisir : 2 puis Entrée**

**Sortie attendue (succès) :**

```
Vérification de la base MySQL (WMS-DB)

Veuillez saisir les identifiants MySQL
Utilisateur MySQL : saisir l'utilisateur
Mot de passe MySQL : Saisir le mot de passe (saisie masquée)
Connexion a MySQL sur 10.5.20.113:3306 ...
- Connexion MySQL : OK
- Requete de test (SELECT 1) : OK

❑ Base MySQL WMS opérationnelle

Appuyez sur Entrée pour revenir au menu...
```

**Sortie en cas d'erreur :**

```
Connexion à 192.168.10.21:3306...
❑ Erreur MySQL : KO

Causes possibles :
- Mot de passe incorrect
- Utilisateur n'existe pas
- Droits insuffisants

Appuyez sur Entrée pour revenir au menu...
```

## 6.5 Option 3 : Informations OS

**Action** : Affiche les informations d'OS des serveurs AD/DNS et la version de MySQL.

**Saisir : 3 puis Entrée**

**Sortie attendue :**

```
=== Informations OS des serveurs ===
```

```
Contrôleurs de domaine AD/DNS:
```

```
DC01 (192.168.10.10) : Windows Server 2019
```

```
DC02 (192.168.10.11) : Windows Server 2019
```

```
Serveur MySQL:
```

```
IP : 192.168.10.21
```

```
OS : Ubuntu 20.04 LTS
```

```
Version MySQL : 8.0.35-0ubuntu0.20.04.1
```

```
Appuyez sur Entrée pour continuer...
```

**Utilité** : Aide à identifier les versions pour suivi obsolescence.

## 6.6 Codes de retour

Le Module 1 retourne un code de sortie exploitable par les scripts :

- **0** : Tous les tests réussis
- **1** : Au moins un test a échoué

**Utilisation en script** :

```
python3 Module1.py
EXIT_CODE=$?

if [ $EXIT_CODE -eq 0 ]; then
    echo "Diagnostic OK"
else
    echo "Problème détecté"
fi
```

---

# 7. Utilisation du Module 2 : Sauvegarde WMS

## 7.1 Lancement du Module 2

Depuis le menu principal, saisir **2** puis **Entrée**.

## 7.2 Configuration interactive

Le Module 2 demande les paramètres de connexion MySQL au lancement :

```
=== Configuration de la connexion MySQL ===  
  
IP serveur MariaDB : 10.5.20.113  
Nom de la base : wms  
Utilisateur : backup  
Mot de passe : ***** (saisie masquée)  
  
Connexion établie avec succès à la base 'wms'.
```

**⚠ Important :** Le mot de passe n'est **pas affiché** lors de la saisie (sécurité).

## 7.3 Menu du Module 2

```
=== Module Sauvegarde WMS ===  
  
1 - Sauvegarde complete SQL  
2 - Export d'une table CSV  
0 - Quitter  
  
Votre choix :
```

## 7.4 Option 1 : Sauvegarde SQL complète

**Action :** Exporte l'intégralité de la base WMS au format SQL restaurable.

**Saisir :** 1 puis **Entrée**

**Déroulement :**

```
=== Sauvegarde SQL complète ===
```

Sauvegarde SQL réussie

**\*\*Fichier généré\*\* :**

- **\*\*Emplacement\*\*** : `backups/wms\_AAAA-MM-JJ\_HH-MM-SS.sql`
- **\*\*Format\*\*** : SQL texte (structure + données)

### 7.5 Option 2 : Export CSV d'une table

**\*\*Action\*\*** : Exporte les données d'une table spécifique au format CSV.

**\*\*Saisir\*\*** : **\*\*2\*\*** puis **\*\*Entrée\*\***

**\*\*Sélection de la table\*\*** :

=== Export table CSV ===

Tables disponibles : 1 commandes 2 produits 3 clients 4 entrepôts 5 lignes\_commande 6 mouvements\_stock 7 utilisateurs 8 rôles 9 fournisseurs 10 catégories 11 emplacement\_stock 12 inventaire 13 expéditions 14 configurations 15 logs\_système 0 Retour

Choisir le numéro de la table : 2

**\*\*Export\*\*** :

Export CSV réussi (produits)

**Fichier généré :**

- **Emplacement** : exports/table\_NomTable\_AAAA-MM-JJ\_HH-MM-SS.csv
- **Format** : CSV avec en-têtes (UTF-8)
- **Séparateur** : Virgule ,

**Exemple de contenu CSV :**

```
id,reference,nom,description,prix_ht,stock_actuel,categorie_id
1,PROD-001,Carton 60x40x40,"Carton standard",2.50,1250,3
2,PROD-002,Palette Europe,"Palette 1200x800",15.00,89,1
3,PROD-003,Film étirable,"Rouleau 500m",8.50,342,2
```

# 8. Utilisation du Module 3 : Audit EOL

## 8.1 Lancement du Module 3

Depuis le menu principal, saisir **3** puis **Entrée**.

**⚠ Important** : Le Module 3 nécessite *nmap* installé et *privileges élevés* (sudo/admin).

**Linux avec sudo :**

```
sudo python3 main.py
# Puis choisir option 3
```

**Windows (Administrateur) :**

- Clic droit sur PowerShell → **Exécuter en tant qu'administrateur**
- Puis lancer `python main.py`

## 8.2 Menu du Module 3

```
MODULE D'AUDIT D'OBSOLESCENCE RÉSEAU
1. Scanner réseau (nmap)
2. Versions OS (local/API - AUTO-MAJ)
3. Analyser CSV (local/API)
4. Quitter

Choix (1-4) :
```

## 8.3 Option 1 : Scanner le réseau

**Action** : Lance un scan nmap pour détecter les hôtes actifs et identifier leurs OS.

**Saisir** : **1** puis **Entrée**

**Paramètres demandés :**

```
Plage (ex: 192.168.1.0/24) : 192.168.10.0/24
```

**Déroulement :**

Scan du réseau 192.168.10.0/24 en cours...  
Ce scan peut prendre plusieurs minutes selon la taille du réseau...

**Durée** : 5 à 15 minutes selon la taille du réseau (/24 = 256 IP).

**Résultat :**

Scan terminé: 8 hôte(s) détecté(s)

IP	OS	Préc.
-----		
192.168.10.10	Microsoft Windows Server 2016	95
192.168.10.11	Microsoft Windows Server 2016	95
192.168.10.21	Ubuntu Linux (nmap: Linux 5.4)	90
192.168.10.22	Ubuntu Linux (nmap: Linux 5.4)	90
192.168.10.40	Linux 3.10 - 4.11	88
192.168.10.50	Linux 4.15 - 5.6	92
192.168.10.100	Microsoft Windows 10 1809	96
192.168.10.101	Microsoft Windows 10 21H2	96

**Sauvegarde CSV :**

CSV? (o/n) : o  
Nom (défaut: scan\_results.csv) : scan\_siege\_2026-02-24.csv  
  
CSV propre sauvé: scan\_siege\_2026-02-24.csv  
Format colonnes: IP | OS | Précision | Ports ouverts

**Fichiers générés :**

- nmap\_scan.xml : Sortie brute nmap (XML)
- scan\_siege\_2026-02-24.csv : Résultats propres (CSV)

## 8.4 Option 2 : Mise à jour données EOL

**Action** : Télécharge les données de fin de support (EOL) depuis l'API endoflife.date.

**Saisir** : 2 puis **Entrée**

**Sous-menu** :

VERSIONS OS EOL

1. Mettre à jour fichier local (API)
2. Local (fichier local - offline)
3. API directe (Internet)
0. Retour

Choisissez (0-3): 1

### Option 1 : Mise à jour locale (recommandé)

Télécharge les données EOL pour 15 OS populaires et les stocke localement :

Mise à jour des données EOL depuis l'API endoflife.date...

Récupération ubuntu... 12 versions  
Récupération debian... 15 versions  
Récupération windows-server... 10 versions  
Récupération windows... 18 versions  
... (15 OS au total)

Mise à jour terminée: 15 OS

Données sauvegardées dans eol\_data\_local.json

**Fichier créé :** eol\_data\_local.json

### Avantages mode local :

- Fonctionne sans Internet
- Rapide (pas de latence réseau)
- Résilient (API indisponible OK)

### Option 2 : Consulter local

Affiche les versions disponibles depuis le cache local pour un OS :



Produit OS (ex: ubuntu): ubuntu

Versions de ubuntu (15 disponibles) :

Version	Sortie	EOL	Statut
24.04	2024-04-25	2034-04-25	Supporté
22.04	2022-04-21	2032-04-21	Supporté
20.04	2020-04-23	2025-04-02	37 jours
18.04	2018-04-26	2023-04-02	EOL dépassé
16.04	2016-04-21	2021-04-02	EOL dépassé

### Option 3 : API directe

Interroge l'API en temps réel (nécessite Internet).

## 8.5 Option 3 : Analyser un scan CSV

**Action** : Analyse le fichier CSV d'un scan nmap et génère un rapport EOL qualifié.

**Saisir** : 3 puis **Entrée**

**Paramètres** :

CSV: scan\_siege\_2026-02-24.csv

Source EOL:

1. Local (offline - AUTO-MAJ si manquant)
2. API (Internet)

Source (1-2): 1

**Rapport final** :

IP	OS	Version	EOL	Statut
192.168.10.10	Microsoft Windows...	2016	2027-01-12	Supporté
192.168.10.11	Microsoft Windows...	2016	2027-01-12	Supporté
192.168.10.21	Ubuntu Linux	5.4	N/A	Inconnu
192.168.10.22	Ubuntu Linux	5.4	N/A	Inconnu
192.168.10.40	Linux 3.10 - 4.11	3.10	N/A	OS inconnu
192.168.10.50	Linux 4.15 - 5.6	4.15	N/A	OS inconnu
192.168.10.100	Microsoft Windows...	1809	2020-11-10	EOL dépassé
192.168.10.101	Microsoft Windows...	21H2	2023-06-13	EOL dépassé

Résumé: 2 EOL dépassé(s), 0 bientôt EOL

Rapport: eol\_audit\_report.csv

Fichier généré : eol\_audit\_report.csv

#### Interprétation :

- **Supporté** : Version en support, plus d'1 an restant
- **EOL dans X jours** : Support expire bientôt (< 1 an)
- **EOL dépassé** : Version hors support (**critique**)
- **Inconnu** : Version non détectée ou OS non mappé

## Annexes

### Annexe A : Ressources externes

#### Documentation officielle :

- Python 3 : <https://docs.python.org/3/> (<https://docs.python.org/3/>)
- nmap : <https://nmap.org/book/man.html> (<https://nmap.org/book/man.html>)
- MySQL : <https://dev.mysql.com/doc/> (<https://dev.mysql.com/doc/>)

#### API EOL :

- endoflife.date : <https://endoflife.date/docs/api> (<https://endoflife.date/docs/api>)

#### Outils recommandés :

- Éditeur Python : VSCode, PyCharm, Sublime Text
- Client MySQL : MySQL Workbench, DBeaver, HeidiSQL
- Analyseur CSV : LibreOffice Calc, Excel, Power BI

---

**Manuel rédigé par le Groupe 2 - B3 ASRBD - EPSI Auxerre**  
**Projet académique MSPR - Année 2025-2026**  
**Février 2026**