

DOCUMENTATION COMPLÈTE

MATELASAPP WESTELYNCK

Version 3.0.1 - 20/07/2025

Système de traitement automatisé des commandes de literie

TABLE DES MATIÈRES

- 1. Vue d'ensemble
- 2. Système de dates et semaines
- 3. Système d'alertes en temps réel
- 4. Modules de création et construction
- 5. Scripts de build manuels
- 6. Installation et configuration
- 7. Utilisation
- 8. Dépannage
- 9. Maintenance

1. VUE D'ENSEMBLE

MatelasApp Westelynck est une application de traitement automatisé de commandes de literie développée pour Westelynck. Elle permet d'analyser des fichiers PDF de commandes, d'extraire les informations de matelas et sommiers, et de générer des fichiers Excel formatés pour la production.

Fonctionnalités principales :

- Analyse PDF : Extraction automatique des données de commandes
- IA/LLM : Enrichissement des données via intelligence artificielle
- Génération Excel : Création de fichiers de production formatés
- Coloration automatique : Mise en forme visuelle des données
- Gestion des semaines : Organisation par semaines de production
- Alertes temps réel : Notifications en temps réel
- Build automatisé : Compilation et déploiement simplifiés

2. SYSTÈME DE DATES ET SEMAINES

Format des semaines de production :

```
S{numéro_semaine}_{année}
```

Exemple: S31_2025 (Semaine 31 de 2025)

Gestion automatique des dates :

- Détection automatique de la semaine courante
- Calcul automatique de la semaine suivante
- Format standardisé : S{WW}_{YYYY}

Exemple de génération de nom de fichier :

```
nom_fichier = f"Matelas_S{numero_semaine}_{annee}_{compteur}.xlsx"
```

Résultat: Matelas_S31_2025_1.xlsx

3. SYSTÈME D'ALERTES EN TEMPS RÉEL

Architecture des alertes :

AlertSystem

■■■ AlertManager (Gestionnaire principal)

■■■ AlertWidget (Interface utilisateur)

■■■ AlertTypes (Types d'alertes)

■■■ AlertQueue (File d'attente)

Types d'alertes disponibles :

- Success : Opérations réussies
- Warning : Avertissements
- Error : Erreurs critiques
- Info : Informations générales
- Progress : Progression des opérations

Caractéristiques des alertes :

- Affichage automatique en temps réel
- Fermeture automatique après délai configurable
- Historique des alertes récentes
- Actions utilisateur (fermer, effacer tout)
- Intégration GUI avec PyQt6

4. MODULES DE CRÉATION ET CONSTRUCTION

Architecture modulaire :

```
MatelasApp/  
■■■ app_gui.py # Interface principale  
■■■ backend/ # Modules backend  
■ ■■■ llm_provider.py # Fournisseurs IA  
■ ■■■ excel_import_utils.py # Utilitaires Excel  
■ ■■■ pdf_utils.py # Traitement PDF  
■ ■■■ date_utils.py # Gestion des dates  
■■■ build_scripts/ # Scripts de build  
■ ■■■ windows/ # Scripts Windows  
■■■ assets/ # Ressources
```

Modules principaux :

1. LLM Provider (backend/llm_provider.py)

- Gestion multi-providers : OpenAI, OpenRouter, Ollama
- Configuration centralisée des clés API
- Fallback automatique en cas d'erreur
- Cache intelligent des réponses

2. Excel Import Utils (backend/excel_import_utils.py)

- Génération Excel avec formatage avancé
- Coloration automatique selon les données
- Alignement intelligent des cellules
- Gestion des templates matelas/sommiers

5. SCRIPTS DE BUILD MANUELS

Scripts ASCII Windows (Recommandés) :

1. Menu Principal (build_scripts/windows/menu_ascii.bat)

```
@echo off
chcp 65001 >nul

:menu
cls
echo =====
echo MATELASAPP - MENU PRINCIPAL
echo =====
echo.
echo Dossier: %CD%
echo.
echo Choisissez une option:
echo.
echo [1] Installation complete (recommandee)
echo [2] Lancer l'application
echo [3] Diagnostic complet
echo [4] Nettoyer les builds
echo [5] Informations
echo [6] Quitter
```

Caractéristiques des scripts ASCII :

- Aucun caractère spécial (émojis, accents, symboles)
- Format ASCII pur compatible Windows
- Syntaxe batch standard sans caractères problématiques
- Encodage UTF-8 avec chcp 65001
- Début correct avec @echo off

6. INSTALLATION ET CONFIGURATION

Prérequis système :

- Python 3.8+ (recommandé 3.11+)
- PyQt6 pour l'interface graphique
- PyInstaller pour la compilation
- Windows 10/11 (pour les scripts batch)

Installation automatique :

```
# Double-cliquer sur  
Lancer_MatelasApp_ASCII.bat  
  
# Ou lancer directement  
build_scripts\windows\menu_ascii.bat
```

Configuration des clés API :

```
{  
  "openrouter": {  
    "api_key": "votre_clé_api_ici",  
    "base_url": "https://openrouter.ai/api/v1"  
  },  
  "openai": {  
    "api_key": "votre_clé_api_ici",  
    "base_url": "https://api.openai.com/v1"  
  }  
}
```


7. UTILISATION

Lancement de l'application :

1. Double-cliquer sur Lancer_MatelasApp_ASCII.bat
2. Choisir l'option 1 pour l'installation complète
3. Attendre la compilation PyInstaller
4. Lancer l'application via l'option 2

Interface utilisateur :

- Panneau gauche : Contrôles et configuration
- Panneau central : Sélection de fichiers
- Panneau droit : Résultats et prévisualisation
- Barre de statut : Informations en temps réel

Traitement des commandes :

1. Sélectionner les fichiers PDF de commandes
2. Configurer le provider LLM (OpenRouter recommandé)
3. Lancer le traitement
4. Vérifier les résultats dans le panneau droit
5. Ouvrir les fichiers Excel générés

8. DÉPANNAGE

Problèmes courants :

1. Erreurs d'encodage Windows

```
'ho' n'est pas reconnu en tant que commande interne
```

Solution : Utiliser les scripts ASCII (*_ascii.bat)

2. Erreurs de clé API

```
401 Client Error: Unauthorized
```

Solution : Vérifier la clé API dans l'interface de gestion

3. Erreurs PyQt6

```
ImportError: cannot import name 'QAction' from 'PyQt6.QtWidgets'
```

Solution : Réinstaller PyQt6 avec `pip install --upgrade PyQt6`

Scripts de diagnostic :

```
# Diagnostic complet
```

```
build_scripts\windows\diagnostic_ascii.bat
```

```
# Test rapide
```

```
test_rapide_ascii.bat
```

```
# Test détaillé
```

```
test_scripts_ascii_windows.bat
```

9. MAINTENANCE

Mise à jour de l'application :

1. Sauvegarder les configurations importantes
2. Télécharger la nouvelle version
3. Remplacer les fichiers (sauf config/)
4. Relancer l'installation

Sauvegarde des données :

- Configuration : config/ (clés API, mappings)
- Référentiels : backend/Référentiels/
- Templates : backend/template/
- Logs : logs/

Monitoring et performance :

- Vérification des logs régulière
- Nettoyage des fichiers temporaires
- Mise à jour des dépendances
- Sauvegarde des configurations

SUPPORT ET CONTACT

Documentation disponible :

- GUIDE_INSTALLATION.md - Guide d'installation
- GUIDE_TEST_SCRIPTS_ASCII.md - Test des scripts
- RESUME_SCRIPTS_ASCII_FINAL.md - Résumé des scripts
- CHANGELOG.md - Historique des versions

En cas de problème :

1. Consulter la documentation
2. Exécuter les scripts de diagnostic
3. Vérifier les logs d'erreur
4. Contacter le support technique

Informations système :

- Version : 3.0.1
- Dernière mise à jour : 19/07/2025
- Compatibilité : Windows 10/11, macOS, Linux
- Langage : Python 3.8+
- Interface : PyQt6

CONCLUSION

MatelasApp Westelynck est une solution complète et robuste pour le traitement automatisé des commandes de literie. Avec son système d'alertes en temps réel, sa gestion intelligente des dates et ses scripts de build optimisés, elle offre une expérience utilisateur fluide et professionnelle.

Les scripts ASCII garantissent une compatibilité maximale avec Windows, éliminant les problèmes d'encodage courants. Le système modulaire permet une maintenance facile et des évolutions futures.

Pour commencer : Double-cliquez sur Lancer_MatelasApp_ASCII.bat et suivez les instructions du menu principal.