

CHAPITRE 32 — Référence des scripts en ligne de commande

Tous les scripts en ligne de commande livrés avec le module. Chemin :
`htdocs/custom/infrasstudio/scripts/`.

rescan_slots.php

Rescane les fichiers tpl.php d'un site et synchronise la table des slots.

```
php rescan_slots.php <website_ref_or_id> [--entity=N] [--purge-orphans] [--dry-run] [--lint]
```

Option	Effet
<code>--entity=N</code>	Force l'entity Dolibarr.
<code>--purge-orphans</code>	Supprime immédiatement les slots orphelins (sans attendre 30 jours).
<code>--dry-run</code>	Affiche les changements sans les appliquer.
<code>--lint</code>	Vérifie la syntaxe des slots. Code de sortie : 0 sans anomalie, 1 avec avertissement, 2 avec erreur.

rebuild_solution_wrappers.php

Reconstruit manuellement les wrappers solution-*.php du catalogue produit.

```
php rebuild_solution_wrappers.php <websitekey-or-id> <public-docroot> [entity]
```

Exemple

```
php rebuild_solution_wrappers.php monsite /var/www/monsite 2
```

generate_sitemap.php

Génère le fichier sitemap.xml d'un site.

```
php generate_sitemap.php <website_ref> [entity]
```

Idéal pour une planification quotidienne via cron :

```
# crontab -e
0 3 * * * php /var/www/dolibarr/htdocs/custom/infrasstudio/scripts/generate_sitemap.php
monsite 2
```

preset_default.php

Chargeur générique de presets JSON. Crée des champs personnalisés produit en série.

```
php preset_default.php <chemin-vers-preset.json>

# Exemple avec le preset livré
php preset_default.php htdocs/custom/infrasstudio/presets/keaticweb.json
```

Format JSON attendu : voir la documentation des presets.

preset_keatic.php

Wrapper de compatibilité — équivalent à `preset_default.php presets/keaticweb.json`.

consolidate_sister_pages.php

Consolide une famille de pages sœurs (modèle B legacy) en une page canonique et plusieurs stubs (modèle A moderne).

```
php consolidate_sister_pages.php <site_ref> \
  [--entity=N] \
  [--base-slug=<slug>] \
  [--dry-run] \
  [--extractor=/path/to/extractor.php]
```

convert_tpl_to_slots.php

Transforme les `echo $langs->trans('Key')` d'un fichier tpl.php en tokens `{{slot:...|default=@lang:Key}}`.

```
php convert_tpl_to_slots.php <tpl_path> [--group=xxx] [--prefix=yyy] [--dry-run] [--hook-engine]
```

Option	Effet
<code>--group=xxx</code>	Attribut group par défaut.
<code>--prefix=yyy</code>	Préfixe des noms de slot.
<code>--dry-run</code>	Aperçu des modifications sans application.
<code>--hook-engine</code>	Ajoute également l'inclusion du moteur du module.

Bonnes pratiques en ligne de commande

- Toujours utiliser `--dry-run` avant une migration importante.
- Lancer en tant qu'utilisateur Apache (`sudo -u www-data php script.php`) pour éviter les problèmes de permissions sur les fichiers générés.
- En multicompany, toujours préciser `--entity=N`.
- Capturer la sortie dans un journal : `php script.php 2>&1 | tee /tmp/script.log`.

Fin de la Partie VI — Vous disposez désormais d'un mémo complet : constantes, shortcodes, hooks, tables SQL, scripts en ligne de commande. Tout est rassemblé en un seul endroit pour une consultation rapide.

Revision #2

Created 2026-05-07 12:19:28 UTC by Lucky Ranasolonirina

Updated 2026-05-08 08:13:12 UTC by Lucky Ranasolonirina