

Kit de documentation

Documentation — Prompts de dialogue — FR

1 Introduction

Un *modèle* de langage applique les règles qu'on lui donne, dans l'ordre où elles ont été données, avec la priorité qu'elles portent. Il n'arbitre pas entre des instructions contradictoires — il les accumule. Ce n'est pas de la mauvaise volonté, c'est de la logique.

Il est également conçu pour être utile et accommodant. Cette combinaison crée un piège classique: l'utilisateur formule une demande qui contredit une règle du Kit, l'*IA* s'exécute aimablement, et la structure se dégrade silencieusement. C'est précisément pour cela que les règles du Kit doivent être posées avant toute demande de production, et que les exceptions non couvertes doivent être traitées comme des évolutions du Kit, jamais comme des dérogations ponctuelles.

Ce document regroupe les commandes à copier-coller dans le *chat* pour maintenir la session dans le cadre du Kit. Quatre situations sont couvertes: le démarrage d'une session de production, le recadrage d'une session déviante, l'autorisation ponctuelle d'une modification chirurgicale, et le contrôle préalable à toute création ou modification de document.

Les blocs de ce document sont donnés en français. Les variantes allemande et anglaise portent les mêmes blocs dans leur langue, et disent la même chose — utiliser celle qui correspond à la langue de travail de la session.

Convention de mise en forme des blocs: le titre est en majuscules, suivi d'une ligne séparatrice de trente signes égal. Les sections internes sont séparées par une ligne de tirets. Les énumérations utilisent des items à tirets, avec continuations alignées sous le texte. Les conditions d'arrêt sont introduites par une flèche ASCII. Chaque nouvelle idée est séparée par une ligne vide contenant un *espace insécable* — sans quoi le convertisseur web supprime la ligne.

2 Commandes de dialogue

2.1 Démarrage de session de production

À utiliser en tête de toute session impliquant la *génération* ou la modification d'un document. Cette commande impose la lecture des sources de vérité avant la première ligne de script.

```
Bonjour. Nouvelle session de production sur le Kit.
Avant toute génération, voici le cadre à respecter.

RÈGLE ABSOLUE — ZÉRO FORMATAGE DE MÉMOIRE
=====

LECTURE OBLIGATOIRE AVANT TOUTE GÉNÉRATION :
-----

1. LIRE le Kit - Projet - Prompt intégralement.
2. LIRE le fichier stylesheet .js actif dans son intégralité
   (General / Glossary / YAML / HTML selon le document
   à produire).
```

3. LIRE le .docx de référence associé – même horodatage que le .js.
§9.1 = format des rows makeTable.
§9.3 = constantes d'indentation et largeurs de colonnes.
§13 = contrats de retour, spread, précurseurs, successeurs.
4. LIRE le prompt du projet courant.
5. LIRE Kit - Documentation - Quality Control §3.13 et §3.14
– chaîne d'appel absolue et prérequis d'environnement.
- Reference .docx – JAMAIS recréé depuis le .js seul.
Le .js contient le code. Le .docx contient la recette d'application : contrats, précurseurs, successeurs, anti-patrons.
Ces informations ne peuvent pas être déduites du .js seul.
Sans le fichier source : arrêt complet.

Aucune valeur numérique, aucune signature de fonction, aucune couleur, aucune constante d'indentation ne peut être utilisée sans avoir été vérifiée dans le .js lu à cette session.

CHAÎNE DE VALIDATION – ORDRE ABSOLU :

0. PRÉREQUIS :
node -e "console.log(Object.keys(require('docx')).length)"
-> nombre supérieur à 200 requis.
node -e "require('adm-zip')" – avant une publication du site seulement ; hors chaîne documentaire depuis General 1.83.
1. AVANT toute génération :
python kit_check_registry.py
-> exit 0 requis.
2. AVANT chaque node gen-*.js :
python kit_check_setters.py gen-document.js
-> exit 0 requis.
3. APRÈS génération .docx :
node kit_validate_docx.js document.docx --version {version}
4. APRÈS génération .xlsx :
python kit_validate_xlsx.py document.xlsx
5. APRÈS régénération d'un document existant :
python kit_check_markup.py source.docx produit.docx
-> un diff de texte ne voit pas une perte de marquage.
6. APRÈS régénération d'une variante de langue :
python kit_check_ossature.py reference.docx variante.docx
7. APRÈS une évolution de l'extraction :
python kit_check_fidelite.py sur le parc, --version {version}
8. AVANT toute livraison touchant un membre de couplet :
python kit_check_couplets.py

Aucun fichier livré tant que la chaîne complète n'a pas passé.

VÉRIFICATIONS AVANT EXÉCUTION DU SCRIPT :

- Le niveau d'un élément de corps vient du titre courant.
h1, h2 et h3 le posent ; les autres helpers le lisent.
Aucun suffixe de niveau à vérifier depuis General 1.70.
- Section du document :
properties: { ...style.pageProps, titlePage: true }
Sans ce drapeau, Word ignore les en-têtes et pieds de page
de première page et la page de garde perd sa nudité.
- makeTable reçoit les cellules au bon format :
chaîne "texte" ou tableau ["texte", italique].
Jamais de largeur numérique à l'intérieur d'une cellule.
- Chargement conditionnel des modules de données selon
[Préfixe] - Registry.requires. Si un flag est false, le
fichier .js correspondant est ABSENT du projet – aucun
require() ni setter. Si true, charger et appeler le setter :
style.setGlossaryTerms(...) / style.setBrands(...)
style.setVariableNames(...) / style.setHassEntities(...)
- style.getLuxTimestamp() utilisé pour l'horodatage.
Jamais réutiliser un horodatage existant.
- Consulter la section Contrats du .docx de référence
(General §13 / YAML §8 / HTML §13) avant tout appel.
- makeImageRun(buffer, largeur, hauteur, format) pour toute image
– format parmi png, jpg, gif, bmp, svg, obligatoire.
- Noms de fichiers sans aucun tiret bas, sauf outils kit_*.
- Couplet intégrité : toute modification d'un fichier d'un
couplet exige mise à jour simultanée du partenaire, même
pour un bump cosmétique.
- Contrôle avant production : demander explicitement ce qui
change ou s'ajoute. Confirmer le nom de fichier exact avant
d'écrire la première ligne.

CONDITIONS D'ARRÊT :

Si une anomalie, incohérence ou ambiguïté est détectée à
n'importe quelle étape :

- > Arrêt complet.
- > Signaler clairement et attendre une confirmation explicite.
- > L'absence de réponse est un signal d'arrêt.
- > Ne jamais signaler et enchaîner dans la même phrase.

Si un cas de formatage n'est pas couvert par le stylesheet :

-> Signaler comme besoin d'évolution du Kit. Ne pas improviser.

Si le document à modifier existe déjà :

-> Relire le fichier source .docx de l'arbre de travail avant de commencer.

-> Repartir de ce fichier, jamais d'un souvenir.

-> Ne modifier que ce qui est explicitement demandé.

2.2 Recadrage — session déviante

À utiliser lorsque la session a dévié au point où les corrections ponctuelles ne suffisent plus. La solution la plus efficace est d'ouvrir un nouveau *chat* et de coller cette commande en premier message.

La session a dérivé. On repart des sources, sans rien reconstruire de mémoire.

RECADRAGE — RETOUR AUX SOURCES

=====

CONTEXTE :

Nouveau chat obligatoire si la session est trop volumineuse.
Purger les chats précédents avant de recommencer.

INFORMATION TECHNIQUE :

- Les noms de fichiers affichés avec des tirets bas sont une transformation de la plateforme. Le nom canonique utilise des espaces.

- Les .docx de l'arbre de travail sont les fichiers sources.
Toute modification repart du fichier lui-même, relu avant d'être touché.

DÉMARRAGE DE SESSION — ORDRE OBLIGATOIRE :

1. Lire le Kit - Projet - Prompt intégralement.
2. Lire les stylesheets .js et leurs .docx de référence.
3. Lire le prompt du projet courant.
4. Lire tous les documents non encore parcourus en détail.
5. Purger tout ce qui contredit ces documents lus.
6. Purger tout ce qui pourrait conduire à une initiative non sollicitée ou à un raccourci quelconque.
7. Rapporter ce qui a été éliminé, puis ce qui reste en mémoire.

RÈGLES DE PRODUCTION :

- Formatage exclusivement selon le stylesheet actif.
Jamais de mélange de stylesheets dans un même document.
Jamais de formatage de mémoire.

- Script générateur recréé from scratch à chaque session.
- Chargement conditionnel des modules de données selon [Préfixe] - Registry.requires. Correspondances :
 - glossary -> Terms.js -> setGlossaryTerms
 - brands -> Brands.js -> setBrands
 - variableNames-> Variable Names -> setVariableNames
 - hassEntities -> HASS Entities -> setHassEntities
- Fichier source .docx relu dans l'arbre de travail avant toute modification.
Ne modifier que ce qui est explicitement demandé.
Aucune perte d'information non autorisée.
- Travailler uniquement depuis l'export le plus récent fourni dans la session. Jamais d'inférence ni de reconstruction depuis un document déjà généré ou depuis la mémoire d'une session précédente.
- Couplet intégrité : toute modification d'un fichier d'un couplet exige mise à jour simultanée du partenaire.
- Signaler toute incohérence avant de lancer la génération du script, et attendre le go explicite.
- Vérifier que le validateur pointe vers la bonne version du fichier.
- Nomenclature stricte : espaces et tirets uniquement, jamais de tirets bas. Horodatage frais à chaque livraison, sans exception.
- Ne jamais poser le niveau autrement qu'en émettant un titre : h1, h2 et h3 sont les seules fonctions qui l'écrivent.

COUPLETS OBLIGATOIRES :

- Stylesheet .js et sa Reference .docx, même horodatage.
- Kit Cover Sheet .docx et .png, même horodatage.
- Cover Sheet projet .docx et .png, même horodatage, indépendant de celui du Kit.
- Module de termes et glossaire .docx, même horodatage.
Un projet multilingue produit un glossaire par langue, tous au même horodatage.
- Cover Sheet : toutes les instructions d'assemblage de l'image se trouvent dans le .docx correspondant. Rendu par constantes uniquement, jamais de composition libre.

2.3 Resserrage — dérive progressive

Les moteurs d'*intelligence artificielle* actuels ont tendance à privilégier ce qui plaît à l'utilisateur sur ce qui lui a été demandé. La dérive est lente et ne déclenche aucune alerte: les réponses s'allongent, des analyses non

demandées s'ajoutent, et des règles pourtant écrites cessent d'être relues. Il faut donc resserrer le moteur régulièrement, sans attendre que la session soit devenue inutilisable.

À utiliser dès que les réponses s'allongent ou s'écartent de la demande. Contrairement au bloc précédent, ce resserrage ne demande pas d'ouvrir un nouveau *chat*: il se colle dans la session en cours.

```
RESSERRAGE
=====

Relis Kit Prompt §1 (checklist) et §2 (ton). Applique-les à
partir de maintenant.

AVANT CHAQUE RÉPONSE :
-----

- Réponds à ce qui est demandé. Rien d'autre.
- Pas d'analyse non demandée, pas de résumé de ce que je
  viens de dire, pas de leçon sur les règles.
- Une valeur de formatage se lit dans le .js et la Reference
  de cette session, jamais de mémoire.
- Si tu ne sais pas, dis-le en une phrase.

Confirme en une ligne, puis continue.
```

2.4 Autorisation de modification chirurgicale

À utiliser uniquement lorsqu'une modification ciblée d'un document existant est nécessaire et qu'une régénération complète serait disproportionnée. À coller juste avant de déposer la demande.

```
Demande précise : modifier un .docx existant par patch
chirurgical, sans le régénérer. Voici les conditions.

RÈGLE — MODIFICATION XML DIRECTE SUR .docx EXISTANT
=====

CONDITIONS REQUISES (toutes simultanément) :
-----

- Le .docx cible a été généré par un script utilisant le
  stylesheet actif. Jamais sur un fichier d'origine inconnue
  ou produit manuellement.

- Le script générateur du document n'est plus disponible dans
  la session. S'il l'est, l'édition XML est interdite : le
  document se régénère.

- Chaque valeur modifiée (couleur, taille, indentation,
  espacement) est tirée du stylesheet .js actif lu dans la
  session courante. Aucune valeur numérique de mémoire.

- La modification est chirurgicale : seul ce qui est
  explicitement demandé est touché. Aucune réécriture,
  reformulation ou ajout non sollicité.
```

- Aucune perte de contenu sur un document existant. Tout ce qui est présent dans le fichier source est intégralement préservé, sauf instruction explicite contraire.
- Avant toute modification : annoncer précisément quels nœuds seront touchés et attendre la confirmation explicite.
- Regex interdit en écriture sur fichier structuré. Toute modification passe par le parseur natif du format. Le regex en lecture pure reste autorisé.
- Pour insérer une note ou un encadré, utiliser les helpers de `kit_patch_helpers.py` – jamais cloner un fragment du document cible.
- Après modification : valider avec `kit_validate_docx.js` et signaler tout avertissement avant de livrer.

CONDITIONS D'ARRÊT :

Si l'une de ces conditions n'est pas remplie :

- > Arrêt complet.
- > Signaler l'anomalie clairement.
- > Attendre une confirmation explicite avant de continuer.
- > Ne jamais signaler et enchaîner dans la même phrase.

2.5 Contrôle préalable — création ou modification

À coller avant toute création ou modification d'un document. Ce contrôle fait déclarer que les sources de vérité ont été lues, que le fichier source a été inventorié, et que tout changement de contenu a été validé avant de toucher au script.

Avant de créer ou de modifier un document, voici le contrôle à passer point par point.

CONTRÔLE PRÉALABLE – CRÉATION / MODIFICATION DOCUMENT

=====

LECTURE OBLIGATOIRE AVANT TOUTE ACTION :

1. LIRE le Kit - Projet - Prompt intégralement.
2. LIRE le stylesheet .js actif dans son intégralité.
3. LIRE le .docx de référence associé (même horodatage).
Ce .docx est la recette d'application du stylesheet – il prime sur toute inférence depuis le code .js.
§9.1 = format des rows makeTable, chaînes et jamais d'objets de cellule.
§9.3 = constantes et largeurs de colonnes.
§13 = contrats de retour, spread, précurseurs, successeurs.
4. LIRE le prompt projet courant.

5. LIRE Kit - Documentation - Quality Control §3.13 et §3.14.

- Reference .docx – JAMAIS recréé depuis le .js seul.
Sans le fichier source : arrêt complet.

RAPPORT DE LECTURE – RÉPONDRE À CHAQUE POINT :

- Kit Projet Prompt lu intégralement : oui / non.
- Stylesheet .js lu intégralement : oui / non, et version.
- Reference .docx lue intégralement : oui / non, et horodatage.
- Prompt projet lu : oui / non.
- Quality Control §3.13 et §3.14 lus : oui / non.

-> Si un point est "non" : arrêt complet. Lire avant de continuer.

SI MODIFICATION D'UN DOCUMENT EXISTANT :

- Relire le fichier source .docx dans l'arbre de travail;
s'il y manque, le demander. Repartir de ce fichier,
jamais d'un souvenir.
- Couplet intégrité : si le fichier modifié fait partie d'un
couplet, le partenaire doit être évalué ET mis à jour dans
la même session. Fichier source partenaire absent :
arrêt complet.
- Inventorier le fichier source : paragraphes, tables, images.
Consigner le décompte exact avant d'écrire le script.
- Extraire les images depuis le fichier source avant tout script.
Aucune image omise sans instruction explicite.
- Le script est recréé from scratch depuis le contenu du
fichier source. Jamais depuis la mémoire ou un script précédent.
- Lister chaque modification de contenu prévue face à la
version source. Attendre la validation explicite avant
d'écrire le script.

CHAÎNE DE VALIDATION :

0. PRÉREQUIS : docx résolvable ; adm-zip seulement avant
une publication du site.
1. AVANT toute génération : python kit_check_registry.py.
2. AVANT node gen-*.js :
python kit_check_setters.py gen-document.js
-> exit 0 requis.
3. APRÈS génération .docx :
node kit_validate_docx.js document.docx --version {version}
4. APRÈS génération .xlsx :
python kit_validate_xlsx.py document.xlsx

5. APRÈS régénération d'un document existant :
python kit_check_markup.py source.docx produit.docx
-> un diff de texte ne voit pas une perte de marquage.
6. APRÈS régénération d'une variante de langue :
python kit_check_ossature.py reference.docx variante.docx
7. APRÈS une évolution de l'extraction :
python kit_check_fidelite.py sur le parc, --version {version}
8. AVANT toute livraison touchant un membre de couplet :
python kit_check_couplets.py

VÉRIFICATION AVANT LIVRAISON DÉFINITIVE :

Comparer le document produit avec l'inventaire du fichier source source :

- Nombre de sections et de headings : identique ?
- Nombre de tableaux : identique ?
- Nombre d'images : identique ?
- Contenu de chaque section : aucune perte ?
- Légendes d'images : présentes et exactes ?

-> Si un écart est détecté : signaler, corriger, revérifier.

-> Livraison uniquement après confirmation explicite zéro perte.

RÈGLES DE FORMATAGE :

- Zéro formatage de mémoire. Toute valeur vérifiée dans le .js ou la Reference lus dans cette session.
- makeTable : rows = chaînes simples ou paires texte et italique. Jamais d'objets de cellule dans rows.
- Largeurs de colonnes : somme égale à la largeur du niveau. Nombre de largeurs égal au nombre de colonnes.
- Niveau de chaque élément cohérent avec le heading parent.
- Section : properties avec titlePage à true.

RÈGLES NOTE ET ENCADRÉ :

- Note : uniquement si l'information est importante, non évidente et absente du corps du texte. Jamais directement sous un heading, jamais sans paragraphe avant. En cas de doute -> paragraphe ordinaire.
- Encadré : uniquement si le conseil apporte ce que le lecteur ne trouve pas dans le texte courant. S'il ne donne rien de nouveau -> supprimer.

CONDITIONS D'ARRÊT :

Si une anomalie, incohérence ou ambiguïté est détectée :

- > Arrêt complet.
- > Signaler clairement et attendre une confirmation explicite.
- > Ne jamais signaler et enchaîner dans la même phrase.

Si un cas de formatage n'est pas couvert par le stylesheet :

- > Signaler comme besoin d'évolution du Kit. Ne pas improviser.