

Kit de documentation

Projet — Convention de nommage

1 Objet

Tous les *fichiers du projet* suivent une convention de nommage unique: fichiers du Kit, fichiers propres au projet, livrables produits en session et exports de données brutes. Un nommage cohérent donne un tri prévisible, une traçabilité des versions et une vérification de *contexte* fiable d'une session à l'autre.

Le projet utilise une convention de nommage unique pour tous les fichiers. Chaque nom de fichier se compose d'un préfixe, d'une catégorie, d'un sujet, d'un tag de langue optionnel, et d'un suffixe horodaté qui enregistre la date de création ou de dernière régénération.

La convention repose sur des principes fondamentaux: espaces au lieu de tirets bas, casse de phrase pour les sujets, traits d'union uniquement dans le segment de date et comme séparateurs entre les segments structurels, et une séparation claire entre préfixe, catégorie, sujet et *horodatage*.

Note: *L'interface du projet **CLAUDE AI** remplace les espaces par des tirets bas lors de l'affichage des noms de fichiers. C'est un comportement de la plateforme — le nom canonique utilise toujours des espaces.*

2 Patron de nommage

2.1 Patron

Patron: Préfixe - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext

Patron étendu pour un document traduit: Préfixe - Catégorie - Sujet - LANG (AAAA-MM-JJ - HHhMM).ext. Le tag de langue est optionnel — voir §2.4.

L'*horodatage* est entre parenthèses et utilise une horloge 24 heures avec la lettre h séparant les heures et les minutes. L'heure est toujours exprimée en heure locale du Luxembourg, CET en hiver et CEST en été.

Note: *Aucun tiret bas nulle part. Les traits d'union séparent les segments structurels. Le sujet utilise la casse de phrase avec espacement naturel.*

Ce patron s'applique à tous les types de fichiers, à l'exception des artefacts stables exécutables documentés en §3.4, outils Unix-style de type `kit_*.py` et `kit_*.js`, et des *générateurs ad hoc* documentés en §3.6. Les feuilles de `style.js` et tous les autres contenus parsés suivent le patron. L'*horodatage* sur les fichiers `.js` permet de vérifier la correspondance entre une *feuille de style* et son document de référence Word.

- **Règle critique:** lorsqu'un fichier est régénéré, l'*horodatage* dans le nom de fichier doit être mis à jour à l'heure actuelle de la régénération. Ne jamais conserver l'ancien *horodatage* sur un fichier dont le contenu a changé.

Note: *Quand plusieurs fichiers liés sont régénérés ensemble, par exemple une feuille de `style.js` et sa référence.docx, ils doivent recevoir le même nouvel horodatage pour rester synchronisés.*

2.2 Segments

- **Préfixe:** identifie à quel périmètre le fichier appartient. Deux préfixes existent dans le système — voir §3.
- **Catégorie:** un regroupement fonctionnel qui contrôle l'ordre de tri alphabétique. Les catégories sont fixes et listées en §7.
- **Sujet:** une description concise du contenu du fichier, écrite en casse de phrase. Les sujets à plusieurs mots utilisent l'espacement naturel.
- **Tag de langue:** optionnel. Code à deux lettres en capitales identifiant la langue de rédaction du document, présent uniquement si celui-ci existe en plusieurs langues — voir §2.4.
- **Horodatage:** la date et l'heure de création ou de dernière régénération, au format entre parenthèses.
- **Extension:** le type de fichier. Les documents de contenu utilisent.docx, les feuilles de style et scripts utilisent.js, les fichiers de données brutes utilisent.txt, les archives de livraison utilisent.zip — §4.5.

2.3 Exemples

Nom de fichier	Préfixe	Catégorie	Sujet
Kit - Stylesheet - General - Code (2026-03-05 - 08h49).js	Kit	<i>Stylesheet</i>	General - Code
Kit - Stylesheet - General - Reference (2026-03-05 - 08h49).docx	Kit	<i>Stylesheet</i>	General - Reference
Kit - Projet - Prompt (2026-03-05 - 08h49).docx	Kit	Projet	<i>Prompt</i>
Kit - Projet - Convention de nommage (2026-03-05 - 08h49).docx	Kit	Projet	Convention de nommage
Arrosage - Project - Prompt (2026-03-05 - 08h49).docx	Arrosage	Project	<i>Prompt</i>

Nom de fichier	Préfixe	Catégorie	Sujet
Arrosage - Documentation - Philosophy - DE (2026-03-05 - 08h49).docx	Arrosage	Documentation	Philosophy, variante DE

2.4 Tag de langue

Un projet peut publier certains de ses documents en plusieurs langues. Le tag de langue distingue les variantes d'un même document sans toucher au reste du nom.

Il se place après le sujet et avant l'*horodatage*, en capitales, parmi les quatre codes supportés: FR, EN, DE, LU.

La décision se prend document par document, jamais projet par projet. Un document qui n'existe que dans une seule langue ne porte aucun tag, même si d'autres documents du même projet sont traduits. Dès qu'un document est traduit, toutes ses variantes portent le tag — y compris celle rédigée dans la langue déclarée au *Registry*. Un fichier nu à côté de trois fichiers taggés rendrait le tri illisible.

- **Catégorie inchangée:** la catégorie et le sujet restent dans la langue du projet, définie par `project.docLanguage`. Seul le tag marque la langue du contenu. Un projet francophone nomme donc sa variante allemande Documentation - Philosophy - DE, jamais Dokumentation - Philosophy - DE.
- **Slug HTML suffixé:** la page publiée porte le même suffixe en minuscules, `philosophy-de.html`. Un document monolingue garde son slug nu, `awtrix.html`. Ces slugs alimentent les liens du sélecteur de langue du *stylesheet* HTML.
- **Une entrée Registry par variante:** chaque langue possède sa propre entrée dans `Registry.documents` avec son *horodatage*. Le *déploiement delta* compare par slug, donc chaque langue se republie indépendamment des autres.
- **Périmètre déclaré dans le générateur:** la liste des langues publiées d'un document se déclare dans le générateur de site du projet, jamais dans le *Registry*. Le périmètre de traduction est une décision éditoriale par document, pas un drapeau stable de projet.

Fichier source	Page publiée	Entrée Registry
Roodt - Documentation - Philosophy - EN (TS).docx	<code>philosophy-en.html</code>	<code>'philosophy-en'</code>

Fichier source	Page publiée	Entrée Registry
Roodt - Documentation - Philosophie - FR (TS).docx	philosophy-fr.html	'philosophy-fr'
Roodt - Documentation - Philosophie - DE (TS).docx	philosophy-de.html	'philosophy-de'
Roodt - Documentation - Awtrix (TS).docx	awtrix.html	'awtrix'

Note: Le glossaire fait exception au patron: un seul module de termes alimente plusieurs documents glossaire, un par langue, tous au même horodatage. Le tag suit la règle générale: `kit_gen_glossaire_docx.js` le pose dès que le Registry porte plus d'une entrée de document glossaire, et l'omet pour un projet monolingue. Le contrat complet est dans Structure commune §6.1. La Cover Sheet, elle, ne se traduit jamais: elle est la première page d'un classeur physique unique, dont la langue est celle du projet.

3 Préfixes

Deux préfixes coexistent dans le système. Ils permettent aux fichiers des deux périmètres de coexister dans la même vue sans ambiguïté.

3.1 Le préfixe Kit

Le préfixe Kit identifie les fichiers d'infrastructure du kit de base: feuilles de style, documents de référence, *prompt* technique, convention de nommage et guide d'initialisation. Ces fichiers ne sont pas modifiés lors du déploiement d'un nouveau projet.

- **Règle de gouvernance:** le préfixe Kit est réservé. L'*IA* ne crée ni ne modifie aucun fichier avec ce préfixe sauf si l'utilisateur a explicitement indiqué qu'il travaille sur une adaptation du kit lui-même. En session normale de projet, tous les nouveaux fichiers portent le préfixe projet.

3.2 Le préfixe projet

Le préfixe projet est choisi lors de l'initialisation du projet. Il est court, sans caractères spéciaux, facile à taper. Tous les fichiers produits pendant les sessions de travail portent ce préfixe.

Projet	Préfixe recommandé
Système d'arrosage	Arrosage
Rénovation cuisine	Cuisine
Documentation réseau	Réseau

3.3 Le préfixe Kit-X

Le préfixe Kit-X identifie les extensions du Kit écrites par son utilisateur: stylesheets supplémentaires, palette de couleurs, registre de ces stylesheets, outils personnels. Ces fichiers ne sont pas remplacés lors d'une mise à jour du Kit, et ils valent pour tous les projets de leur auteur sans être renommés. Le X est pour extension.

- **Il ne commence pas par "Kit -"**: le séparateur diffère — Kit-X, sans espaces autour du tiret. La règle de gouvernance du §3.1 tient donc sans exception: un fichier "Kit -" appartient au Kit et sera remplacé, un fichier "Kit-X -" appartient à l'utilisateur.
- **Fichiers connus à ce jour**: Kit-X - *Registry.js*, Kit-X - Colors (TS).js, Kit-X - *Stylesheet* - [Sujet] - Code (TS).js et sa Reference. La liste n'est pas fermée: un utilisateur nomme ainsi tout ce qu'il ajoute au Kit.
- **Distinction d'avec le préfixe projet**: Kit-X pour ce qui suit son auteur d'un projet à l'autre, le préfixe du projet pour ce qui ne vaut que pour lui. Recette d'écriture: *Étendre le Kit*.

3.4 Artefacts stables — nomenclature snake_case

Les artefacts stables exécutables — outils Unix-style qui vivent en permanence dans l'arbre, Kit ou projet, et dont la logique reste stable session après session — suivent une convention distincte du patron général. Ils portent le préfixe du projet en minuscules suivi d'un tiret bas, kit_ pour les outils du Kit de base et [préfixe]_ pour ceux d'un *projet consommateur*, utilisent snake_case pour le nom, et ne portent pas d'*horodatage*. Exception formelle à la règle d'interdiction des tirets bas énoncée en §2.

Exemples conformes Kit: kit_validate_docx.js, kit_validate_xlsx.py, kit_render_cover_sheet.py, kit_check_setters.py, kit_check_markup.py, kit_check_ossature.py, kit_patch_helpers.py, kit_gen_glossaire_html.js. Exemples projet: trading_generate_history_report.py.

Contrat opérationnel complet: *Quality Control* §3. Recette de création pour *projets consommateurs*: *Structure commune* §12. Critère de distinction: un *artefact stable* exécutable suit la convention snake_case; tout autre fichier — *feuille de style*, *module de données*, *registry*, document, *générateur ad hoc*, fichier source — suit le patron §2 ou §3.6 selon le cas.

3.5 Slug d'un document de moteur

Un document propre à un moteur d'*intelligence artificielle* porte le préfixe de ce moteur dans son slug, en minuscules: claudemanuel-fr, grok-prompts-de-dialogue-de, chatgpt-guide-pratique-en. Un slug nu appartient au Kit lui-même, dont les documents ne dépendent d'aucun moteur.

Le slug est une adresse publiée. Le renommer casse les liens déjà partagés: la règle s'applique dès la création du document, et un renommage se décide comme une rupture, non comme un rangement. kit_check_registry.py

signale un document de moteur dont le slug ne porte pas son préfixe, et un slug préfixé d'un moteur pour un document qui n'est pas le sien.

3.6 Générateurs ad hoc — patron gen-*

Les *générateurs ad hoc* sont des scripts **NODE.JS** recréés from scratch à chaque session pour produire un livrable.docx ou un site HTML déployable. Contrairement aux artefacts stables de §3.4, ils ne vivent pas en permanence dans l'*arbre du projet* — ils sont jetables, leur durée de vie est celle de la session courante.

Patron: gen-{slug-document}.js. Exemples: gen-cover-sheet.js, gen-pipeline-html.js, gen-{prefix}-site.js. Pas de préfixe projet redondant: chaque projet vit dans son propre dossier de session.

Exception nommée: gen-kit-site.js pour le Kit et gen-{prefix}-site.js pour un *projet consommateur* — le suffixe -site identifie sans ambiguïté le générateur de *pipeline* HTML du *sous-site*, distinct des générateurs de documents.docx.

Note: *Pas d'horodatage dans le nom du générateur — son TS est celui de l'artefact qu'il produit, lu via getLuxTimestamp() dans le générateur lui-même. Convention alignée sur la pratique du Kit et sur la cohérence projet: chaque session repart from scratch, l'historique des générateurs n'est pas conservé.*

Exemples corrects: gen-arch-threat-model.js, gen-build-versioning.js, gen-ops-continuity.js, gen-cover-sheet.js, gen-pipeline-html.js, gen-sliver-app-site.js.

Anti-patrons: préfixe projet redondant, par exemple gen-sliver-app-prompt-v2.js là où gen-prompt-v2.js suffit. Mélange de préfixe court et de projet complet dans la même session — choisir une convention et s'y tenir.

4 Noms des livrables de session

En plus des fichiers projet principaux, l'*IA* produit des livrables de session qui utilisent un patron date-et-version. Ces noms apparaissent dans l'*inventaire* des fichiers projet et permettent de suivre l'évolution des documents entre les sessions.

4.1 Fichiers de collecte de données

Patron: AAAA-MM-JJ - Collecte de données - [SOURCE] - [description].txt

Le système source apparaît comme deuxième espace réservé, permettant à tous les fichiers de données de se trier ensemble par ordre alphabétique sous "Collecte de données".

Exemples: 2026-03-05 - Collecte de données - HASS - automations yaml.txt et 2026-03-05 - Collecte de données - UniFi - export réseau.txt.

4.2 Fichiers de script

Patron: AAAA-MM-JJ - Script - *Python* - [nom].py

Les scripts *Python* utilisés pour la collecte de données sont rangés dans l'*arbre du projet*.

4.3 Inventaire des fichiers projet

Patron: AAAA-MM-JJ - *Inventaire* fichiers projet - Version 1.xx.txt

L'*inventaire* est un fichier texte brut listant chaque fichier devant être présent dans l'*arbre du projet*. Il est produit comme dernière étape de chaque session et est auto-référent.

4.4 Résumé de session

Patron: [Préfixe] - Projet - Résumé de session (AAAA-MM-JJ - HHhMM).docx

Le résumé de session est généré à la fin de chaque session sous forme de document Word formaté selon la *feuille de style* générale. Il comporte cinq sections: travail accompli, décisions prises, travail en attente, questions en suspens et recommandations. Contrairement aux autres livrables de session, il suit le patron principal §2.1.

4.5 Archives de livraison

Une session livre son travail en archives, jamais fichier par fichier — *Structure commune* §11. Elles ne suivent pas toutes la même règle: la plupart nomment ce qu'elles contiennent, l'archive de téléchargement nomme une adresse.

Archive	Patron	Origine du nom
Extension Kit-X	Kit-X - [Sujet] (TS).zip	Le préfixe Kit-X, qui suit son auteur
Projet complet	[Préfixe] - Projet complet (TS).zip	Le préfixe du projet, comme tout livrable
Site publié	{siteName} (TS).zip	Registry.deploy.siteName
Passe incrémentale	{siteName} - delta (TS).zip	Registry.deploy.siteName
Téléchargement publié	{filename}, sans <i>horodatage</i>	Registry.projectZip.filename

Une archive de déploiement porte le nom du domaine où elle va, non celui du projet qui la produit: elle se reconnaît à sa destination. Un projet qui publie plusieurs *sous-sites* en produit autant, distinguées par ce nom.

Note: *L'archive de téléchargement est la seule sans horodatage. Son nom est une adresse — le lien de la page d'accueil pointe dessus — et une adresse qui change à chaque passe est une adresse morte. Elle porte donc un numéro de version, incrémenté sur demande explicite. L'archive d'une extension Kit-X se livre séparément de celle du projet: elle ne se range pas au même endroit, puisqu'elle suit son auteur d'un projet à l'autre — Structure commune §3 et §11.*

Note: *“Projet complet” et “delta” sont des littéraux invariants, non des sujets localisables: seuls [Préfixe], {siteName}, {filename} et (TS) se substituent. Une archive n'est pas un*

document — ni page de garde, ni lecteur, ni traduction. Son nom sert au tri d'une sauvegarde, où la continuité chronologique d'une même famille compte davantage que la langue: un projet qui bascule sa langue de documentation ne doit pas voir ses archives changer de nom et perdre leur historique. Le §5 ne s'applique donc pas ici.

5 Noms localisés

Les catégories et sujets de nom de fichier sont localisés selon la langue du projet, définie par le champ `project.docLanguage` dans Kit - *Registry.js* pour le Kit, ou [Préfixe] - *Registry.js* pour un *projet consommateur*. Quatre langues sont supportées: EN, défaut pour les *projets consommateurs*, FR, langue historique du Kit, DE et LU.

Règle de cohérence: au sein d'un même projet, toutes les catégories et tous les sujets de ses fichiers suivent la langue déclarée. Pas de mélange de langues dans les noms de fichiers d'un même projet. Le Kit lui-même reste en FR par exception documentée.

Cette règle porte sur les segments du nom, pas sur la langue de rédaction du document. Un projet francophone qui traduit un document en allemand conserve ses catégories françaises et marque la langue du contenu par le tag de §2.4.

5.1 Catégories localisées

Table de référence des catégories fixes traduites dans les quatre langues supportées.

Catégorie (EN)	FR	DE	LU
Annex	Annexe	Anhang	Annex
Documentation	Documentation	Dokumentation	Dokumentatioun
Glossary	Glossaire	Glossar	Glossär
Manual	Manuel	Handbuch	Handbuch
Project	Projet	Projekt	Projet
Stylesheet	<i>Stylesheet</i>	<i>Stylesheet</i>	<i>Stylesheet</i>

Note: La catégorie *Stylesheet* reste identique dans les quatre langues — terme technique anglais à valeur universelle, aligné sur la terminologie CSS et Word.

5.2 Sujets de documents Kit localisés

Table de référence pour les noms de documents Kit publiés. Les sujets qui ne figurent pas dans cette table, noms techniques stables comme *Cover Sheet*, *Pipeline HTML*, *Quality Control* ou *Reading Guide*, restent en anglais dans toutes les langues. Les noms d'archive sortent du champ de cette table: ils sont invariants — §4.5.

Sujet (EN)	FR	DE	LU
Prompt	<i>Prompt</i>	<i>Prompt</i>	<i>Prompt</i>
Naming conventions	Convention de nommage	Namenskonventionen	Nummkonventionen

Sujet (EN)	FR	DE	LU
Initialization guide	Guide d'initialisation	Initialisierungsleitfaden	Initialisierungsguide
Update guide	Guide de mise à jour	Aktualisierungsleitfaden	Aktualisierungsguide
Common structure	Structure commune	Gemeinsame Struktur	Gemeinsam Struktur
Todos	Todos	Todos	Todos
Manual	Manuel	Handbuch	Handbuch
Practical guide	Guide pratique	Praktischer Leitfaden	Praktesche Guide
Session summary	Résumé de session	Sitzungszusammenfassung	Sitzungszesummefassung
Dialogue prompts	Prompts de dialogue	Dialogprompts	Dialogprompts
Glossary	Glossaire	Glossar	Glossär

5.3 Exemples de nom complet par langue

Langue	Exemple Kit	Exemple projet
EN	Kit - Project - <i>Prompt</i> (2026-04-23 - 14h30).docx	Trading - Project - <i>Prompt</i> (2026-04-23 - 14h30).docx
FR	Kit - Projet - <i>Prompt</i> (2026-04-23 - 14h30).docx	Trading - Projet - <i>Prompt</i> (2026-04-23 - 14h30).docx
DE	Kit - Projekt - <i>Prompt</i> (2026-04-23 - 14h30).docx	Trading - Projekt - <i>Prompt</i> (2026-04-23 - 14h30).docx
LU	Kit - Projet - <i>Prompt</i> (2026-04-23 - 14h30).docx	Trading - Projet - <i>Prompt</i> (2026-04-23 - 14h30).docx

Ces exemples illustrent la *localisation* des segments du nom. Pour un document existant en plusieurs langues, le tag de §2.4 s'ajoute après le sujet: Trading - Projet - *Prompt* - DE (2026-04-23 - 14h30).docx dans un projet francophone.

5.4 Renommage lors d'un changement de langue

Quand `project.docLanguage` est modifié pour un projet actif, tous les *fichiers du projet* doivent être renommés pour refléter les catégories et

sujets de la nouvelle langue. L'*horodatage* des fichiers est également rafraîchi à cette occasion puisque la régénération est complète. Procédure détaillée: *Guide de mise à jour*.

Note: *Le Kit lui-même ne change pas de langue — `project.docLanguage` vaut FR dans Kit - Registry.js, verrouillé par exception documentée dans Prompt. Les projets consommateurs débutent à EN par défaut.*

6 Numérotation des versions

Tous les fichiers versionnés utilisent le patron x.yy où x est la version majeure et yy la version mineure. La version majeure n'est incrémentée que pour les refontes structurelles. La version mineure est incrémentée à chaque mise à jour.

Suffixes de révision itérative: pendant une session, lorsque l'*IA* produit plusieurs révisions d'un document en réponse aux retours de l'utilisateur, chaque fichier téléchargeable ajoute un suffixe alphabétique minuscule au numéro de version, par exemple 1.01a, 1.01b, 1.01c. Le suffixe est uniquement dans le nom de fichier — la version interne du document reste au numéro de base.

Note: *Les suffixes de révision aident l'utilisateur à identifier quel téléchargement est lequel pendant un cycle itératif. Ils sont supprimés lorsque la version finale est confirmée.*

7 Catégories

Les catégories sont des étiquettes fixes qui forment le deuxième segment du nom de fichier. Elles contrôlent l'ordre de tri alphabétique dans le dossier projet et regroupent les fichiers associés. La table ci-dessous utilise les noms anglais — équivalents dans les autres langues en §5.1.

Catégorie	Préfixe	Fonction
Annexe	Projet	Documents PDF de référence déposés manuellement — rapports, fiches techniques, docs tiers. Chaque PDF est couplé à un .docx Kit court qui résume son contenu
Documentation	Kit	Documents de référence transversaux: reading guide, <i>cover sheet</i> , <i>pipeline</i> HTML, quality control, prompts de dialogue
Glossary	Projet	Document glossaire, liste de termes, modules de formatage

Catégorie	Préfixe	Fonction
Manuel	Projet	Documents PDF originaux côté constructeur ou producteur — manuels d'équipement, datasheets. Chaque PDF est couplé à un .docx Kit court décrivant le matériel et l'usage
Project	Kit ou Projet	Fichiers de gestion de projet: <i>prompt</i> , convention de nommage, guide d'initialisation, guide de mise à jour, structure commune, résumés de session, <i>inventaire</i>
Stylesheet	Kit	Modules de <i>feuille de style</i> .js et leurs références Word .docx — toujours préfixés Kit

Note: Les catégories ne sont pas créées de manière ad hoc. L'ajout d'une nouvelle catégorie est une décision de projet qui doit être enregistrée dans le *prompt* projet.

Note: Pour les fichiers Stylesheet, le sujet est composé de deux parties séparées par un trait d'union: le qualificateur — General, Glossary, YAML, HTML — et le type, Code pour les fichiers .js et Reference pour les fichiers .docx.

8 Référence rapide

Contexte	Patron	Exemple
Fichier Kit	Kit - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext	Kit - Projet - <i>Prompt</i> (2026-03-05 - 08h49).docx
Fichier projet	[Préfixe] - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext	Arrosage - Project - <i>Prompt</i> (2026-03-05 - 08h49).docx
Document traduit	[Préfixe] - Catégorie - Sujet - LANG (AAAA-MM-JJ - HHhMM).ext	Roodt - Documentation - Philosophy - DE (2026-08-17 - 13h55).docx
Couplet annexe ou manuel	[Préfixe] - {Annexe Manuel} - [Nom] (TS).docx et [Préfixe] - {Annexe Manuel} - [Nom].pdf	Sliver - Annexe - Schéma électrique (2026-05-06 - 14h00).docx
Résumé de session	[Préfixe] - Projet - Résumé de session (AAAA-MM-JJ - HHhMM).docx	Arrosage - Project - Session summary (2026-03-05 - 21h45).docx

Contexte	Patron	Exemple
Export de données	AAAA-MM-JJ - Collecte de données - [SOURCE] - [description].txt	2026-03-05 - Collecte de données - HASS - export.txt
Script	AAAA-MM-JJ - Script - <i>Python</i> - [nom].py	2026-03-05 - Script - <i>Python</i> - collect.py
Inventaire	AAAA-MM-JJ - <i>Inventaire</i> fichiers projet - Version 1.xx.txt	2026-03-05 - <i>Inventaire</i> fichiers projet - Version 1.01.txt
Artefact Kit	kit_sujet.ext	kit_check_markup.py
Artefact projet	[préfixe]_sujet.ext	trading_generate_history_report.py
Générateur ad hoc	gen-{slug-document}.js	gen-pipeline-html.js, gen-cover-sheet.js
Générateur HTML site	gen-{prefix}-site.js	gen-kit-site.js, gen-trading-site.js