Kit de documentation

Projet — Convention de nommage

1 Introduction

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 pour tout document, et d'un suffixe horodaté qui enregistre la date de création ou de dernière regénération; les exceptions sont nommées au §2.1.

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 d'un document: Préfixe - Catégorie - Sujet - [Langue du document] (AAAA-MM-JJ - HHhMM).ext

Patron d'un fichier sans langue de contenu — *feuille de style*, *module de données*, script: Préfixe - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext. Tag de langue: §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 §3.3, outils Unix-style de type kit_*.py et kit_*.js, et des *générateurs ad hoc* documentés en §3.5. 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.

- **Sans catégorie ni horodatage:** [Préfixe] - *Registry.js*, source des horodatages, et [Préfixe] - Settings.js — *Structure commune* §5.4.
- **Sans catégorie, avec horodatage:** les *modules de données* Brands, Text Highlights, Document Titles, Colors, Variable Friendly Names et HASS Entities, par exemple Kit - Brands (TS).js.

- **Nom imposé par l'outil qui le lit:** package.json et package-lock.json, que *npm* lit sous ce nom, et les icônes du site — favicon.svg, favicon.ico, apple-touch-icon.png, index-icon.svg.
- **Sans horodatage:** la pièce d'une annexe ou d'un manuel, [Préfixe] - {Annexe|Manuel} - [Nom].pdf — §8 et *Pipeline HTML* §6.3.
- **Photo liée, sans horodatage:** assets/png/{slug}/{slug}-{section}-{n}.jpg ou .png. Le document la désigne par ce nom: une photo remplacée garde le sien — *General Reference* §13.5, *HTML Reference* §14.7. Sa variante dans une langue, pour une traduction, porte le même nom suivi de la langue en minuscules: {slug}-{section}-{n}-de.jpg.
- **Date en tête:** les livrables de session — exports de données, scripts, inventaires — §4.1 à §4.3.
- **Archives:** l'archive du site et celle du projet suivent leur propre patron — §4.5.
- **Règle critique:** lorsqu'un fichier est regénéré, l'*horodatage* dans le nom de fichier doit être mis à jour à l'heure actuelle de la regénération. Ne jamais conserver l'ancien *horodatage* sur un fichier dont le contenu a changé, hors l'image de couverture rendue seule, qui prend celui de son .docx — *Prompt* §14.

Note: *Quand plusieurs fichiers liés sont regénérés 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. Les préfixes du système sont décrits en §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:** code à deux lettres en capitales identifiant la langue de rédaction du document. Tout document le porte, qu'il existe en une langue ou en plusieurs — voir §2.4.
- **Horodatage:** la date et l'heure de création ou de dernière regénération, 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 - 08 h49).js	Kit	*Stylesheet*	General - Code
Kit - *Stylesheet* - General - Reference - FR (2026-03-05 - 08 h49).docx	Kit	*Stylesheet*	General - Reference
Kit - Projet - Prompt - FR (2026-03-05 - 08 h49).docx	Kit	Projet	*Prompt*
Kit - Projet - Convention de nommage - FR (2026-03-05 - 08 h49).docx	Kit	Projet	Convention de nommage
[Préfixe] - Project - Prompt - EN (2026-03-05 - 08h49).docx	[Préfixe]	Project	*Prompt*
Kit - Documentation - Reading Guide - DE (2026-03-05 - 08h49).docx	Kit	Documentation	Reading Guide, variante DE

2.4 Tag de langue

Le tag de langue dit dans quelle langue un document est rédigé. Il distingue aussi les variantes d'un même document publié en plusieurs langues, 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. Dans un patron, il s'écrit [Langue du document] — [Dokumentsprache] et [Document language] dans les variantes allemande et anglaise d'un document du Kit —, et en minuscules dans un slug: comment-documenter-[langue du document].html.

Tout document porte le tag, qu'il existe en une seule langue ou en plusieurs. Un document traduit plus tard garde ainsi le nom de sa variante d'origine: la traduction s'ajoute à côté, sans renommage. Le tag se lit aussi sans ouvrir le fichier, et le

lecteur d'une autre langue sait d'emblée dans quelle langue le document est rédigé. Un fichier sans langue de contenu — *feuille de style*, *module de données*, *Registry*, script — n'en porte pas.

- **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. Le Kit, projet francophone, nomme donc la variante allemande de son prompt *Kit - Projet - Prompt - DE*, jamais Kit - Projekt - Prompt - DE.
- **Slug HTML suffixé:** la page publiée d'un document traduit porte le même suffixe en minuscules, reading-guide-de.html. Un document publié dans une seule langue garde son slug nu, sans suffixe, bien que son fichier porte le tag: le slug est une adresse publiée, et il ne prend son suffixe qu'à la traduction du document — §3.4. 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*. La clé de l'entrée est le slug: celle d'un document publié dans une seule langue reste nue. 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 se décide document par document. Cette règle porte sur le document. Le périmètre du site — les langues qu'il sert, qu'un document soit traduit ou non — est un drapeau stable, et se déclare au *Registry* sous project.siteLanguages: *Pipeline HTML* §5.

Fichier source	Page publiée	Entrée Registry
Kit - Documentation - Reading Guide - FR (TS).docx	reading-guide-fr.html	'reading-guide-fr'
Kit - Documentation - Reading Guide - DE (TS).docx	reading-guide-de.html	'reading-guide-de'
Kit - Documentation - Reading Guide - EN (TS).docx	reading-guide-en.html	'reading-guide-en'
[Préfixe] - Documentation - [Sujet] - EN (TS).docx	[slug].html	'[slug]'

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 sur chaque document glossaire, projet monolingue compris. Le contrat complet est dans Structure commune §6.1. La Cover Sheet reste un document unique, rédigé dans la langue du projet dont elle porte le tag; ses rendus .png se produisent*

par langue publiée, et une page de suite insère " - Page N" avant l'horodatage — Cover Sheet §4.

3 Préfixes

Chaque préfixe désigne un périmètre: le Kit et ses moteurs, ou le projet. Les fichiers de périmètres différents coexistent ainsi dans la même vue sans ambiguïté.

3.1 Le préfixe Kit

Le préfixe Kit identifie les fichiers d'infrastructure du Kit: 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. Les préfixes de moteur — **CLAUDE AI**, et celui de tout moteur qui rejoint le Kit — sont des préfixes du Kit: ils nomment ses documents propres à un moteur d'*intelligence artificielle*, et la règle de gouvernance ci-dessous les couvre — slug en §3.4.

- **Règle de gouvernance:** la règle et son exception vivent au *Kit Prompt* §7. 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
[Nom du projet]	[Préfixe]

3.3 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. Un projet nomme les siens de même, sous son préfixe.

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, hors gen-kit-site.js et gen-glossary-reference.js, qui gardent leur nom — §3.5; tout autre fichier —

feuille de style, *module de données*, *registry*, document, *générateur ad hoc*, fichier source — suit le patron §2 ou §3.5 selon le cas.

3.4 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: claude-manuel-fr, claude-guide-pratique-de, claude-environnement-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.

3.5 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. Contrairement aux artefacts stables de §3.3, ils ne vivent pas en permanence dans l'*arbre du projet* — ils sont jetables, leur durée de vie est celle de la session courante. Un document existant se régénère, lui, depuis sa carte par kit_gen_document.js — *Quality Control* §3.16.

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

Exception: le générateur de site du Kit d'un projet est un fichier projet durable. C'est le gen-*-site.js qui requiert kit_publication.js, un seul par projet; gen-[préfixe]-site.js en est le nom recommandé, non réservé. Un projet peut porter à côté un autre générateur de site, hors Kit, qui ne requiert pas kit_publication.js: le Kit ne le lit ni ne le contrôle. Il s'écrit une fois par projet, vit dans l'arbre et dans l'archive, et s'entretient ensuite: sans lui, l'archive d'un *sous-site*, reprise, ne produit pas son site — *Publication et reprise* §8, *Structure commune* §2. Le suffixe -site le distingue des générateurs de documents. gen-kit-site.js et gen-glossary-reference.js sont, eux, des artefacts stables du Kit, livrés avec lui et jamais réécrits en session — *Quality Control* §3. Ils gardent leur nom, sous lequel documents et outils les appellent.

Note: *Pas d'horodatage dans le nom d'un 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 de documents n'est pas conservé.*

Exemples corrects: gen-cover-sheet.js, gen-pipeline-html.js, gen-structure-commune.js, et pour le site d'un projet gen-[préfixe]-site.js.

Anti-patrons: préfixe projet redondant, par exemple gen-[préfixe]-prompt.js là où gen-prompt.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 - *pandoc* - version.txt et 2026-03-05 - Collecte de données - *npm* - modules.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 fait partie des livrables de fin de session, produits sur le go de l’utilisateur — *Prompt* §10.2 —, et est auto-référent.

4.4 Résumé de session

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

Le résumé de session est un document Word formaté selon la *feuille de style* générale, produit en fin de session sur le go de l’utilisateur — *Prompt* §10.2; sa structure est celle du *Prompt* §10.3. 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. Leurs noms suivent des règles distinctes: l’archive du site publié porte le nom du site et un *horodatage*, quelle que soit l’étendue de la passe, puisqu’elle contient toujours le site entier; l’archive du projet porte un numéro de version.

Archive	Patron	Origine du nom
Site publié	{siteName} (TS).zip	Registry.deploy.siteName
Archive du projet	{nom}-{version}.zip, sans *horodatage*	Registry.projectZip.filename, qui finit par -{version}.zip, version étant projectZip.version

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 du projet est la seule sans horodatage. Tout projet la produit: c'est elle que l'utilisateur garde et redonne au fil suivant. Partagée, son nom est aussi 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, qui monte à chaque livraison. Une archive déjà produite ne se refait jamais sous le même numéro: absente au départ d'une passe, elle se demande — Structure commune §11.*

Note: *Les segments fixes des patrons restent les mêmes dans toutes les langues: seuls {siteName}, {nom}, {version} et (TS) se substituent. Une archive est un contenant de fichiers. 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*. Les langues prises en charge sont EN, défaut pour les *projets consommateurs*, FR, langue historique du Kit, DE et LU.

Règle de cohérence: au sein d'un même projet, les catégories et sujets des documents suivent la langue déclarée, sans mélange. Les fichiers sans langue de contenu — feuilles de style, *modules de données*, *Registry* — gardent leur nom technique anglais dans toutes les langues: générateurs et outils les cherchent sous ce nom. 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 chaque langue du Kit.

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	*Stylesheet*	*Stylesheet*	*Stylesheet*

Note: *La catégorie Stylesheet reste identique dans toutes les langues — terme technique anglais à*

valeur universelle, aligné sur la terminologie CSS et Word.

5.2 Sujet et libellé des documents Kit

Les libellés localisés des documents du Kit ont leur source unique dans Kit - Document Titles (TS).js, dont titleEntries porte le libellé de chaque document dans chaque langue — *Structure commune* §6.8. Le sujet d'un nom de fichier et le libellé d'un texte se distinguent. Dans un nom de fichier, le sujet reste celui de la langue du projet dans toutes les variantes — §2.4 —, et les noms techniques stables, comme *Cover Sheet*, *Pipeline* HTML, *Quality Control* ou Reading Guide, y restent invariants. Dans un texte, une page ou un renvoi, un document se nomme par son libellé dans la langue du texte, tel que Document Titles le porte. Les noms d'archive sont invariants — §4.5.

5.3 Exemples de nom complet par langue

Langue du projet	Exemple
EN	[Préfixe] - Project - Prompt - EN (2026-04-23 - 14h30).docx
FR	Kit - Projet - Prompt - FR (2026-04-23 - 14h30).docx
DE	[Préfixe] - Projekt - Prompt - DE (2026-04-23 - 14h30).docx
LU	[Préfixe] - Projet - Prompt - LU (2026-04-23 - 14h30).docx

Ces exemples illustrent la *localisation* des segments du nom. Les segments suivent la langue du projet, le tag de §2.4 celle du contenu: le Kit, projet de langue FR, nomme la variante allemande de son prompt Kit - Projet - Prompt - DE (2026-04-23 - 14h30).docx.

5.4 Renommage lors d'un changement de langue

Quand project.docLanguage change pour un projet actif, tous les *fichiers du projet* sont renommés selon les catégories et sujets de la nouvelle langue, et tous les documents sont régénérés dans cette langue, avec un *horodatage* frais. Le changement se fait en un seul lot, par l'*IA*, sur demande de l'utilisateur — *Guide de mise à jour* §6.

Note: *Le Kit lui-même ne change pas de langue: il est rédigé en français, et project.docLanguage vaut FR dans Kit - Registry.js — Kit Prompt §17. Les projets consommateurs débutent à EN par défaut.*

6 Numérotation des versions

Ce qui porte une version dans son nom ou au *Registry* — feuilles de style, archive d'un projet, archive du Kit, inventaire des fichiers projet, §4.3 — utilise le patron x.yy, où x est la version majeure et yy la version mineure. La version mineure est incrémentée à chaque

mise à jour; celle d'une archive l'est une fois par archive livrée, non à chaque fichier modifié — *Structure commune* §11. La version majeure est incrémentée pour une refonte structurelle, et après x.99, sans refonte: vient alors x+1.00. Les outils kit_ et ceux d'un projet, [préfixe]_, se tiennent par le changelog daté en tête du fichier — *Quality Control* §3, *Structure commune* §12; leurs entrées nouvelles ne portent que leur date.

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
Annex	Projet	Documents PDF de référence déposés manuellement — rapports, fiches techniques, docs tiers. Chaque PDF est accompagné d'un .docx Kit court qui résume son contenu
Documentation	Kit ou projet	Documents de référence transversaux: reading guide, *cover sheet*, *pipeline* HTML, quality control, prompts de dialogue
Glossary	Kit ou projet	Document glossaire, liste de termes, modules de formatage
Manual	Projet	Documents PDF originaux côté constructeur ou producteur — manuels d'équipement, datasheets. Chaque PDF est accompagné d'un .docx Kit court décrivant le matériel et l'usage
Project	Kit ou projet	Fichiers de gestion de projet: prompt, convention de nommage, guide d'initialisation, guide de mise à jour, structure commune, résumés de session, inventaire
Stylesheet	Kit ou projet	Modules de *feuille de style* .js et leurs références Word .docx

Note: *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. La Reference, document, porte le tag de langue; le Code n'en porte pas.*

8 Référence rapide

Contexte	Patron	Exemple
Document Kit	Kit - Catégorie - Sujet - [Langue du document] (AAAA-MM-JJ - HHhMM).ext	Kit - Projet - Prompt - FR (2026-03-05 - 08h49).docx
Document projet	[Préfixe] - Catégorie - Sujet - [Langue du document] (AAAA-MM-JJ - HHhMM).ext	[Préfixe] - Project - Prompt - EN (2026-03-05 - 08h49).docx
Document traduit	[Préfixe] - Catégorie - Sujet - [Langue du document] (AAAA-MM-JJ - HHhMM).ext	Kit - Documentation - Reading Guide - DE (2026-08-17 - 13h55).docx
Fichier sans langue	[Préfixe] - Catégorie - Sujet (AAAA-MM-JJ - HHhMM).ext	Kit - *Stylesheet* - General - Code (2026-03-05 - 08h49).js
Fiche et pièce d'annexe ou de manuel	[Préfixe] - {Annexe|Manuel} - [Nom] - [Langue du document] (TS).docx et [Préfixe] - {Annexe|Manuel} - [Nom].pdf	[Préfixe] - Annexe - [Nom] - FR (2026-05-06 - 14h00).docx
Résumé de session	[Préfixe] - Projet - Résumé de session - [Langue du document] (AAAA-MM-JJ - HHhMM).docx	[Préfixe] - Project - Session summary - EN (2026-03-05 - 21h45).docx
Export de données	AAAA-MM-JJ - Collecte de données - [SOURCE] - [description].txt	2026-03-05 - Collecte de données - *pandoc* - version.txt
Script	AAAA-MM-JJ - Script - *Python* - [nom].py	2026-03-05 - Script - *Python* - collect.py
Inventaire	AAAA-MM-JJ - Inventaire fichiers projet - Version 1.xx.txt	2026-03-05 - Inventaire fichiers projet - Version 1.01.txt
***Artefact* Kit**	kit_sujet.ext	kit_check_markup.py
***Artefact* projet**	[préfixe]_sujet.ext	[préfixe]_validate_[sujet].py
Générateur ad hoc	gen-{slug-document}.js	gen-pipeline-html.js, gen-cover-sheet.js
Générateur HTML site	gen-[préfixe]-site.js	gen-[préfixe]-site.js