Dokumentations-Kit

Projekt — Namenskonvention

1 Einleitung

Alle *Projektdateien* folgen einer einheitlichen Benennungskonvention: Dateien des Kits, projekteigene Dateien, in der Sitzung erzeugte Lieferungen und Rohdatenexporte. Eine stimmige Benennung ergibt eine vorhersehbare Sortierung, eine Rückverfolgbarkeit der Versionen und eine verlässliche Kontextprüfung von einer Sitzung zur nächsten.

Das Projekt verwendet für alle Dateien eine einheitliche Benennungskonvention. Jeder Dateiname besteht aus einem Präfix, einer Kategorie, einem Thema, einem Sprachkürzel bei jedem Dokument und einem *Zeitstempel*-Suffix, das das Datum der Erstellung oder der letzten Neuerzeugung festhält; die Ausnahmen nennt §2.1.

Die Konvention beruht auf grundlegenden Prinzipien: Leerzeichen statt Unterstrichen, Satzschreibweise für die Themen, Bindestriche nur im Datumssegment und als Trenner zwischen den strukturellen Segmenten, sowie eine klare Trennung zwischen Präfix, Kategorie, Thema und *Zeitstempel*.

Hinweis: *Die Oberfläche des **Claude AI**-Projekts ersetzt bei der Anzeige von Dateinamen die Leerzeichen durch Unterstriche. Das ist ein Verhalten der Plattform — der kanonische Name verwendet stets Leerzeichen.*

2 Benennungsmuster

2.1 Muster

Muster eines Dokuments: Präfix - Kategorie - Thema - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).ext

Muster einer Datei ohne Inhaltssprache — *Stylesheet*, *Datenmodul*, Skript: Präfix - Kategorie - Thema (JJJJ-MM-TT - HHhMM).ext. Sprachkürzel: §2.4.

Der *Zeitstempel* steht in Klammern und verwendet eine 24-Stunden-Uhr mit dem Buchstaben h als Trenner zwischen Stunden und Minuten. Die Uhrzeit wird stets in Luxemburger Ortszeit angegeben, MEZ im Winter und MESZ im Sommer.

Hinweis: *Nirgends Unterstriche. Die Bindestriche trennen die strukturellen Segmente. Das Thema verwendet die Satzschreibweise mit natürlichen Leerzeichen.*

Dieses Muster gilt für alle Dateitypen, mit Ausnahme der in §3.3 dokumentierten stabilen ausführbaren Artefakte, Werkzeuge im Unix-Stil vom Typ kit_*.py und kit_*.js, sowie der in §3.5 dokumentierten *Ad-hoc-Generatoren*. Die Stylesheets .js und alle übrigen geparsten Inhalte folgen dem Muster. Der *Zeitstempel* auf den .js-Dateien erlaubt es, die Zuordnung zwischen einem *Stylesheet* und seinem Word-Referenzdokument zu prüfen.

- **Ohne Kategorie und ohne Zeitstempel:** [Präfix] - *Registry.js*, Quelle der *Zeitstempel*, und [Präfix] - Settings.js — *Gemeinsame Struktur* §5.4.
- **Ohne Kategorie, mit Zeitstempel:** die *Datenmodule* Brands, Text Highlights, Document Titles, Colors, Variable Friendly Names und HASS Entities, zum Beispiel Kit - Brands (TS).js.

- **Name vom lesenden Werkzeug vorgegeben:** package.json und package-lock.json, die *npm* unter diesem Namen liest, sowie die Icons der Website — favicon.svg, favicon.ico, apple-touch-icon.png, index-icon.svg.
- **Ohne Zeitstempel:** das Stück eines Anhangs oder Handbuchs, [Präfix] - {Annexe|Manuel} - [Name].pdf — §8 und *HTML-Pipeline* §6.3.
- **Verknüpftes Foto, ohne Zeitstempel:** assets/png/{slug}/{slug}-{section}-{n}.jpg oder .png. Das Dokument bezeichnet es mit diesem Namen: ein ersetztes Foto behält ihn — *General Reference* §13.5, *HTML Reference* §14.7. Seine Variante in einer Sprache, für eine Übersetzung, trägt denselben Namen, gefolgt von der Sprache in Kleinbuchstaben: {slug}-{section}-{n}-de.jpg.
- **Datum vorne:** die Sitzungslieferungen — Datenexporte, Skripte, Dateiverzeichnisse — §4.1 bis §4.3.
- **Archive:** das Archiv der Website und das des Projekts folgen ihrem eigenen Muster — §4.5.
- **Kritische Regel:** wird eine Datei neu erzeugt, muss der *Zeitstempel* im Dateinamen auf die aktuelle Uhrzeit der Neuerzeugung gesetzt werden. Nie den alten *Zeitstempel* auf einer Datei belassen, deren Inhalt sich geändert hat, außer beim allein gerenderten Deckblattbild, das den seiner .docx übernimmt — *Prompt* §14.

Hinweis: *Werden mehrere zusammengehörige Dateien gemeinsam neu erzeugt, zum Beispiel ein Stylesheet .js und seine Reference .docx, müssen sie denselben neuen Zeitstempel erhalten, um synchron zu bleiben.*

2.2 Segmente

- **Präfix:** gibt an, zu welchem Geltungsbereich die Datei gehört. Die Präfixe des Systems sind in §3 beschrieben.
- **Kategorie:** eine funktionale Gruppierung, die die alphabetische Sortierreihenfolge steuert. Die Kategorien sind fest und in §7 aufgeführt.
- **Thema:** eine knappe Beschreibung des Dateiinhalts, in Satzschreibweise geschrieben. Themen aus mehreren Wörtern verwenden natürliche Leerzeichen.
- **Sprachkürzel:** zweibuchstabiger Code in Großbuchstaben, der die Abfassungssprache des Dokuments angibt. Jedes Dokument trägt ihn, ob es in einer Sprache existiert oder in mehreren — siehe §2.4.
- **Zeitstempel:** Datum und Uhrzeit der Erstellung oder der letzten Neuerzeugung, im Klammerformat.
- **Erweiterung:** der Dateityp. Inhaltsdokumente verwenden .docx, Stylesheets und Skripte verwenden .js, Rohdatendateien verwenden .txt, Lieferarchive verwenden .zip — §4.5.

2.3 Beispiele

Dateiname	Präfix	Kategorie	Thema
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
[Präfix] - Project - Prompt - EN (2026-03-05 - 08h49).docx	[Präfix]	Project	*Prompt*
Kit - Documentation - Reading Guide - DE (2026-03-05 - 08h49).docx	Kit	Documentation	Reading Guide, Variante DE

2.4 Sprachkürzel

Das Sprachkürzel gibt an, in welcher Sprache ein Dokument abgefasst ist. Es unterscheidet zudem die Varianten desselben in mehreren Sprachen veröffentlichten Dokuments, ohne den Rest des Namens anzutasten.

Es steht nach dem Thema und vor dem *Zeitstempel*, in Großbuchstaben, als einer der vier unterstützten Codes: FR, EN, DE, LU. In einem Muster wird es [Dokumentsprache] geschrieben — [Langue du document] und [Document language] in der französischen und englischen Fassung eines Kit-Dokuments —, und in Kleinbuchstaben in einem Slug: comment-documenter-[dokumentsprache].html.

Jedes Dokument trägt das Kürzel, ob es in einer einzigen Sprache existiert oder in mehreren. Ein später übersetztes Dokument behält so den Namen seiner

Ursprungsvariante: die Übersetzung kommt daneben hinzu, ohne Umbenennung. Das Kürzel ist zudem lesbar, ohne die Datei zu öffnen, und der Leser einer anderen Sprache weiß sofort, in welcher Sprache das Dokument abgefasst ist. Eine Datei ohne Inhaltssprache — *Stylesheet*, *Datenmodul*, *Registry*, Skript — trägt keines.

- **Kategorie unverändert:** Kategorie und Thema bleiben in der Sprache des Projekts, festgelegt durch project.docLanguage. Allein das Kürzel kennzeichnet die Sprache des Inhalts. Das Kit, ein französischsprachiges Projekt, benennt die deutsche Variante seines Prompts daher *Kit - Projet - Prompt - DE*, nie Kit - Projekt - Prompt - DE.
- **HTML-Slug mit Suffix:** die veröffentlichte Seite eines übersetzten Dokuments trägt dasselbe Suffix in Kleinbuchstaben, reading-guide-de.html. Ein nur in einer Sprache veröffentlichtes Dokument behält seinen nackten Slug, ohne Suffix, obwohl seine Datei das Kürzel trägt: der Slug ist eine veröffentlichte Adresse und erhält sein Suffix erst mit der Übersetzung des Dokuments — §3.4. Diese Slugs speisen die Links der Sprachauswahl des HTML-Stylesheets.
- **Ein Registry-Eintrag je Variante:** jede Sprache besitzt ihren eigenen Eintrag in Registry.documents mit ihrem *Zeitstempel*. Der Schlüssel des Eintrags ist der Slug: der eines nur in einer Sprache veröffentlichten Dokuments bleibt nackt. Die *Delta-Veröffentlichung* vergleicht nach Slug, daher wird jede Sprache unabhängig von den anderen neu veröffentlicht.
- **Umfang im Generator deklariert:** die Liste der veröffentlichten Sprachen eines Dokuments wird im Site-Generator des Projekts deklariert, nie in der *Registry*. Der Übersetzungsumfang wird Dokument für Dokument entschieden. Diese Regel betrifft das Dokument. Der Umfang der Website — die Sprachen, die sie bedient, ob ein Dokument übersetzt ist oder nicht — ist ein stabiler Schalter und wird in der *Registry* unter project.siteLanguages deklariert: *HTML-Pipeline* §5.

Quelldatei	Veröffentlichte Seite	Registry-Eintrag
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'
[Präfix] - Documentation - [Thema] - EN (TS).docx	[slug].html	'[slug]'

Hinweis: *Das Glossar bildet eine Ausnahme vom Muster: ein einziges Begriffsmodul speist mehrere*

Glossardokumente, eines je Sprache, alle mit demselben Zeitstempel. Das Kürzel folgt der allgemeinen Regel: kit_gen_glossaire_docx.js setzt es auf jedes Glossardokument, einsprachige Projekte eingeschlossen. Der vollständige Vertrag steht in Gemeinsame Struktur §6.1. Das Cover Sheet bleibt ein einziges Dokument, abgefasst in der Sprache des Projekts, deren Kürzel es trägt; seine .png-Renderings entstehen je veröffentlichter Sprache, und eine Folgeseite fügt " - Page N" vor dem Zeitstempel ein — Deckblatt §4.

3 Präfixe

Jedes Präfix bezeichnet einen Geltungsbereich: das Kit und seine Engines, oder das Projekt. Dateien unterschiedlicher Geltungsbereiche bestehen so in derselben Ansicht ohne Mehrdeutigkeit nebeneinander.

3.1 Das Präfix Kit

Das Präfix Kit kennzeichnet die Infrastrukturdateien des Kits: Stylesheets, Referenzdokumente, technischer Prompt, *Namenskonvention* und *Einrichtungsleitfaden*. Diese Dateien werden bei der Einrichtung eines neuen Projekts nicht verändert. Die Engine-Präfixe — **CLAUDE AI** und das jeder Engine, die dem Kit beitritt — sind Präfixe des Kits: sie benennen seine Dokumente, die einer Engine der künstlichen Intelligenz eigen sind, und die nachstehende Governance-Regel erfasst sie — Slug in §3.4.

- **Governance-Regel:** die Regel und ihre Ausnahme stehen in *Kit Prompt* §7. In einer normalen Projektsitzung tragen alle neuen Dateien das Projektpräfix.

3.2 Das Projektpräfix

Das Projektpräfix wird bei der Initialisierung des Projekts gewählt. Es ist kurz, ohne Sonderzeichen und leicht zu tippen. Alle während der Sitzungen erzeugten Dateien tragen dieses Präfix.

Projekt	Empfohlenes Präfix
[Projektname]	[Präfix]

3.3 Stabile Artefakte — Nomenklatur snake_case

Die stabilen ausführbaren Artefakte — Werkzeuge im Unix-Stil, die dauerhaft im *Arbeitsbaum* leben, Kit oder Projekt, und deren Logik Sitzung um Sitzung stabil bleibt — folgen einer vom allgemeinen Muster abweichenden Konvention. Sie tragen das Projektpräfix in Kleinbuchstaben, gefolgt von einem Unterstrich, kit_ für die Werkzeuge des Basis-Kits und [präfix]_ für die eines Anwenderprojekts, verwenden snake_case für den Namen und tragen keinen *Zeitstempel*. Formelle Ausnahme von der in §2 ausgesprochenen Regel des Unterstrich-Verbots.

Konforme Beispiele 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. Ein Projekt benennt seine eigenen ebenso, unter seinem Präfix.

Vollständiger Betriebsvertrag: *Qualitätssicherung* §3. Erstellungsrezept für *Anwenderprojekte*: *Gemeinsame Struktur* §12. Unterscheidungskriterium: ein stabiles ausführbares *Artefakt* folgt der Konvention snake_case, außer gen-kit-site.js und gen-glossary-reference.js, die ihren Namen behalten — §3.5; jede andere Datei — *Stylesheet*, *Datenmodul*, *Registry*, Dokument, *Ad-hoc-Generator*, Quelldatei — folgt je nach Fall dem Muster §2 oder §3.5.

3.4 Slug eines Engine-Dokuments

Ein Dokument, das einer Engine der künstlichen Intelligenz eigen ist, trägt das Präfix dieser Engine in seinem Slug, in Kleinbuchstaben: claude-manuel-fr, claude-guide-pratique-de, claude-environnement-en. Ein nackter Slug gehört dem Kit selbst, dessen Dokumente von keiner Engine abhängen.

Der Slug ist eine veröffentlichte Adresse. Ihn umzubenennen bricht bereits geteilte Links: die Regel gilt ab der Erstellung des Dokuments, und eine Umbenennung wird als Bruch entschieden, nicht als Aufräumen. kit_check_registry.py meldet ein Engine-Dokument, dessen Slug nicht sein Präfix trägt, sowie einen mit einer Engine präfixierten Slug für ein Dokument, das nicht zu ihr gehört.

3.5 Ad-hoc-Generatoren — Muster gen-*

Die *Ad-hoc-Generatoren* sind **Node.js**-Skripte, die in jeder Sitzung from scratch neu angelegt werden, um eine .docx-Lieferung zu erzeugen. Anders als die stabilen Artefakte aus §3.3 leben sie nicht dauerhaft im *Arbeitsbaum* des Projekts — sie sind Einwegskripte, ihre Lebensdauer ist die der laufenden Sitzung. Ein bestehendes Dokument wird dagegen aus seiner Karte durch kit_gen_document.js neu erzeugt — *Qualitätssicherung* §3.16.

Muster: gen-{slug-document}.js. Beispiele: gen-cover-sheet.js, gen-pipeline-html.js. Kein redundantes Projektpräfix: jedes Projekt lebt in seinem eigenen Sitzungsordner.

Ausnahme: der Kit-Website-Generator eines Projekts ist eine dauerhafte Projektdatei. Es ist die gen-*-site.js, die kit_publication.js einbindet, eine je Projekt; gen-[präfix]-site.js ist ihr empfohlener, nicht reservierter Name. Ein Projekt kann daneben einen weiteren Website-Generator außerhalb des Kits führen, der kit_publication.js nicht einbindet: das Kit liest und prüft ihn nicht. Er wird einmal je Projekt geschrieben, lebt im *Arbeitsbaum* und im Archiv und wird anschließend gepflegt: ohne ihn erzeugt ein übernommenes *Subsite*-Archiv seine Website nicht — *Veröffentlichung und Übernahme* §8, *Gemeinsame Struktur* §2. Das Suffix -site unterscheidet ihn von den Dokumentgeneratoren. gen-kit-site.js und gen-glossary-reference.js sind dagegen stabile Artefakte des Kits, mit ihm ausgeliefert und nie in einer Sitzung neu geschrieben — *Qualitätssicherung* §3. Sie behalten ihren Namen, unter dem Dokumente und Werkzeuge sie aufrufen.

Hinweis: *Kein Zeitstempel im Namen eines Generators — sein TS ist der des Artefakts, das er*

erzeugt, gelesen über getLuxTimestamp() im Generator selbst. Konvention im Einklang mit der Praxis des Kits und mit der Projektstimmigkeit: jede Sitzung beginnt from scratch, die Historie der Dokumentgeneratoren wird nicht aufbewahrt.

Korrekte Beispiele: gen-cover-sheet.js, gen-pipeline-html.js, gen-structure-commune.js, und für die Website eines Projekts gen-[präfix]-site.js.

Anti-Patterns: redundantes Projektpräfix, zum Beispiel gen-[präfix]-prompt.js, wo gen-prompt.js genügt. Vermischung von kurzem Präfix und vollständigem Projektnamen in derselben Sitzung — eine Konvention wählen und dabei bleiben.

4 Namen der Sitzungslieferungen

Zusätzlich zu den Hauptdateien des Projekts erzeugt die *KI* Sitzungslieferungen, die ein Muster aus Datum und Version verwenden. Diese Namen erscheinen im Dateiverzeichnis des Projekts und erlauben es, die Entwicklung der Dokumente zwischen den Sitzungen zu verfolgen.

4.1 Dateien der Datenerhebung

Muster: JJJJ-MM-TT - Collecte de données - [SOURCE] - [Beschreibung].txt

Das Quellsystem erscheint als zweiter Platzhalter, sodass sich alle Datendateien alphabetisch gemeinsam unter “Collecte de données” einsortieren.

Beispiele: 2026-03-05 - Collecte de données - *pandoc* - version.txt und 2026-03-05 - Collecte de données - *npm* - modules.txt.

4.2 Skriptdateien

Muster: JJJJ-MM-TT - Script - *Python* - [name].py

Die für die Datenerhebung verwendeten *Python*-Skripte werden im *Arbeitsbaum* des Projekts abgelegt.

4.3 Dateiverzeichnis des Projekts

Muster: JJJJ-MM-TT - Inventaire fichiers projet - Version 1.xx.txt

Das Dateiverzeichnis ist eine reine Textdatei, die jede Datei auflistet, die im *Arbeitsbaum* des Projekts vorhanden sein muss. Es gehört zu den Lieferungen zum Sitzungsende, die auf das Go des Nutzers erzeugt werden — *Prompt* §10.2 —, und ist selbstreferenziell.

4.4 Sitzungszusammenfassung

Muster: [Präfix] - Projet - Résumé de session - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).docx

Die Sitzungszusammenfassung ist ein Word-Dokument, formatiert nach dem allgemeinen *Stylesheet* und am Ende der Sitzung auf das Go des Nutzers erzeugt — *Prompt* §10.2; ihr Aufbau ist der aus *Prompt* §10.3. Anders als die übrigen Sitzungslieferungen folgt sie dem Hauptmuster §2.1.

4.5 Lieferarchive

Eine Sitzung liefert ihre Arbeit in Archiven, nie Datei für Datei — *Gemeinsame Struktur* §11. Ihre Namen folgen verschiedenen Regeln: das Archiv der veröffentlichten Website trägt den Namen der Website und einen *Zeitstempel*, unabhängig vom Umfang des Durchlaufs, da es stets die gesamte Website enthält; das Projektarchiv trägt eine Versionsnummer.

Archiv	Muster	Herkunft des Namens
Veröffentlichte Website	{siteName} (TS).zip	Registry.deploy.siteName
Projektarchiv	{Name}-{version}.zip, ohne *Zeitstempel*	Registry.projectZip.filename, das auf -{version}.zip endet, wobei version projectZip.version ist

Ein Deployment-Archiv trägt den Namen der Domain, für die es bestimmt ist, nicht den des Projekts, das es erzeugt: man erkennt es an seinem Ziel. Ein Projekt, das mehrere *Subsites* veröffentlicht, erzeugt ebenso viele, unterschieden durch diesen Namen.

Hinweis: *Das Projektarchiv ist das einzige ohne Zeitstempel. Jedes Projekt erzeugt es: dieses Archiv bewahrt der Nutzer auf und gibt es dem nächsten Arbeitsfaden mit. Geteilt, ist sein Name zudem eine Adresse — der Link der Startseite zeigt darauf —, und eine Adresse, die sich bei jedem Durchlauf ändert, ist eine tote Adresse. Es trägt daher eine Versionsnummer, die mit jeder Lieferung steigt. Ein bereits erzeugtes Archiv wird nie unter derselben Nummer neu erstellt: fehlt es zu Beginn eines Durchlaufs, wird es angefordert — Gemeinsame Struktur §11.*

Hinweis: *Die festen Segmente der Muster bleiben in allen Sprachen gleich: nur {siteName}, {Name}, {version} und (TS) werden ersetzt. Ein Archiv ist ein Behälter von Dateien. Sein Name dient der Sortierung einer Sicherung, in der die chronologische Kontinuität einer Familie mehr zählt als die Sprache: ein Projekt, das seine Dokumentationssprache umstellt, darf seine Archive nicht umbenannt sehen und ihre Historie verlieren. §5 gilt hier daher nicht.*

5 Lokalisierte Namen

Die Kategorien und Themen der Dateinamen werden nach der Sprache des Projekts lokalisiert, festgelegt durch das Feld project.docLanguage in Kit - *Registry.js* für das Kit oder [Präfix] - *Registry.js* für ein *Anwenderprojekt*. Die unterstützten Sprachen sind EN, Vorgabe für *Anwenderprojekte*, FR, historische Sprache des Kits, DE und LU.

Stimmigkeitsregel: innerhalb eines Projekts folgen die Kategorien und Themen der Dokumente der deklarierten Sprache, ohne Vermischung. Die Dateien ohne Inhaltssprache — Stylesheets, *Datenmodule*, *Registry* — behalten in allen Sprachen ihren englischen technischen Namen: Generatoren und Werkzeuge suchen sie unter diesem Namen. Das Kit selbst bleibt als dokumentierte Ausnahme in FR.

Diese Regel betrifft die Segmente des Namens, nicht die Abfassungssprache des Dokuments. Ein französischsprachiges Projekt, das ein Dokument ins Deutsche übersetzt,

behält seine französischen Kategorien und kennzeichnet die Sprache des Inhalts durch das Kürzel aus §2.4.

5.1 Lokalisierte Kategorien

Referenztabelle der festen Kategorien, übersetzt in jede Sprache des Kits.

Kategorie (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*

Hinweis: *Die Kategorie Stylesheet bleibt in allen Sprachen gleich — englischer Fachbegriff mit universeller Geltung, im Einklang mit der Terminologie von CSS und Word.*

5.2 Thema und Bezeichnung der Kit-Dokumente

Die lokalisierten Bezeichnungen der Kit-Dokumente haben ihre einzige Quelle in Kit - Document Titles (TS).js, dessen titleEntries die Bezeichnung jedes Dokuments in jeder Sprache führt — *Gemeinsame Struktur* §6.8. Das Thema eines Dateinamens und die Bezeichnung in einem Text sind zu unterscheiden. Im Dateinamen bleibt das Thema in allen Varianten das der Projektsprache — §2.4 —, und die stabilen technischen Namen, wie *Cover Sheet*, *Pipeline* HTML, *Quality Control* oder Reading Guide, bleiben dort unverändert. In einem Text, einer Seite oder einem Verweis wird ein Dokument mit seiner Bezeichnung in der Sprache des Textes genannt, wie Document Titles sie führt. Die Archivnamen sind unveränderlich — §4.5.

5.3 Beispiele vollständiger Namen je Sprache

Sprache des Projekts	Beispiel
EN	[Präfix] - Project - Prompt - EN (2026-04-23 - 14h30).docx
FR	Kit - Projet - Prompt - FR (2026-04-23 - 14h30).docx
DE	[Präfix] - Projekt - Prompt - DE (2026-04-23 - 14h30).docx
LU	[Präfix] - Projet - Prompt - LU (2026-04-23 - 14h30).docx

Diese Beispiele veranschaulichen die *Lokalisierung* der Namenssegmente. Die Segmente folgen der Sprache des Projekts, das Kürzel aus §2.4 der des Inhalts:

das Kit, ein Projekt der Sprache FR, benennt die deutsche Variante seines Prompts Kit - Projet - Prompt - DE (2026-04-23 - 14h30).docx.

5.4 Umbenennung bei einem Sprachwechsel

Ändert sich project.docLanguage für ein aktives Projekt, werden alle *Projektdateien* nach den Kategorien und Themen der neuen Sprache umbenannt, und alle Dokumente werden in dieser Sprache neu erzeugt, mit frischem *Zeitstempel*. Der Wechsel erfolgt in einem einzigen Durchgang, durch die *KI*, auf Anfrage des Anwenders — *Aktualisierungsleitfaden* §6.

Hinweis: *Das Kit selbst wechselt seine Sprache nicht: es ist auf Französisch abgefasst, und project.docLanguage hat in Kit - Registry.js den Wert FR — Kit Prompt §17. Anwenderprojekte starten standardmäßig auf EN.*

6 Versionsnummerierung

Was im Namen oder im *Registry* eine Version trägt — Stylesheets, Projektarchiv, Kit-Archiv, Dateiverzeichnis des Projekts, §4.3 — verwendet das Muster x.yy, wobei x die Hauptversion und yy die Nebenversion ist. Die Nebenversion wird bei jeder Aktualisierung erhöht; die eines Archivs einmal je geliefertem Archiv, nicht mit jeder geänderten Datei — *Gemeinsame Struktur* §11. Die Hauptversion wird bei einer strukturellen Überarbeitung erhöht, und nach x.99 auch ohne Überarbeitung: dann folgt x+1.00. Die Werkzeuge kit_ und die eines Projekts, [präfix]_, werden über das datierte Changelog am Kopf der Datei geführt — *Qualitätssicherung* §3, *Gemeinsame Struktur* §12; ihre neuen Einträge tragen nur ihr Datum.

7 Kategorien

Die Kategorien sind feste Bezeichnungen, die das zweite Segment des Dateinamens bilden. Sie steuern die alphabetische Sortierreihenfolge im Projektordner und gruppieren die zusammengehörigen Dateien. Die nachstehende Tabelle verwendet die englischen Namen — Entsprechungen in den anderen Sprachen in §5.1.

Kategorie	Präfix	Funktion
Annex	Projekt	Manuell abgelegte PDF-Referenzdokumente — Berichte, technische Datenblätter, Dokumente Dritter. Jedes PDF wird von einer kurzen Kit-.docx begleitet, die seinen Inhalt zusammenfasst
Documentation	Kit oder Projekt	Übergreifende Referenzdokumente: Leseleitfaden, *Deckblatt*, HTML-*Pipeline*, Qualitätssicherung, Gesprächsbefehle

Kategorie	Präfix	Funktion
Glossary	Kit oder Projekt	Glossardokument, Begriffsliste, Formatierungsmodule
Manual	Projekt	Original-PDF-Dokumente von Hersteller- oder Produzentenseite — Gerätehandbücher, Datenblätter. Jedes PDF wird von einer kurzen Kit-.docx begleitet, die das Gerät und seine Verwendung beschreibt
Project	Kit oder Projekt	Dateien der Projektverwaltung: Prompt, Namenskonvention, Einrichtungsleitfaden, Aktualisierungsleitfaden, Gemeinsame Struktur, Sitzungszusammenfassungen, Dateiverzeichnis
Stylesheet	Kit oder Projekt	*Stylesheet*-Module .js und ihre Word-Referenzen .docx

Hinweis: *Das Hinzufügen einer neuen Kategorie ist eine Projektentscheidung, die im Projekt-Prompt festgehalten werden muss.*

Hinweis: *Bei den Stylesheet-Dateien besteht das Thema aus zwei durch einen Bindestrich getrennten Teilen: dem Qualifikator — General, Glossary, YAML, HTML — und dem Typ, Code für die .js-Dateien und Reference für die .docx-Dateien. Die Reference, ein Dokument, trägt das Sprachkürzel; der Code trägt keines.*

8 Kurzreferenz

Kontext	Muster	Beispiel
Kit-Dokument	Kit - Kategorie - Thema - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).ext	Kit - Projet - Prompt - FR (2026-03-05 - 08h49).docx
Projektdokument	[Präfix] - Kategorie - Thema - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).ext	[Präfix] - Project - Prompt - EN (2026-03-05 - 08h49).docx
Übersetztes Dokument	[Präfix] - Kategorie - Thema - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).ext	Kit - Documentation - Reading Guide - DE (2026-08-17 - 13h55).docx

Kontext	Muster	Beispiel
Datei ohne Sprache	[Präfix] - Kategorie - Thema (JJJJ-MM-TT - HHhMM).ext	Kit - *Stylesheet* - General - Code (2026-03-05 - 08h49).js
Merkblatt und Stück eines Anhangs oder Handbuchs	[Präfix] - {Annexe|Manuel} - [Name] - [Dokumentsprache] (TS).docx und [Präfix] - {Annexe|Manuel} - [Name].pdf	[Präfix] - Annexe - [Name] - FR (2026-05-06 - 14h00).docx
Sitzungszusammenfassung	[Präfix] - Projet - Résumé de session - [Dokumentsprache] (JJJJ-MM-TT - HHhMM).docx	[Präfix] - Project - Session summary - EN (2026-03-05 - 21h45).docx
Datenexport	JJJJ-MM-TT - Collecte de données - [SOURCE] - [Beschreibung].txt	2026-03-05 - Collecte de données - *pandoc* - version.txt
Skript	JJJJ-MM-TT - Script - *Python* - [name].py	2026-03-05 - Script - *Python* - collect.py
Dateiverzeichnis	JJJJ-MM-TT - Inventaire fichiers projet - Version 1.xx.txt	2026-03-05 - Inventaire fichiers projet - Version 1.01.txt
Kit-*Artefakt*	kit_thema.ext	kit_check_markup.py
Projektartefakt	[präfix]_thema.ext	[präfix]_validate_[thema].py
Ad-hoc-Generator	gen-{slug-document}.js	gen-pipeline-html.js, gen-cover-sheet.js
HTML-Site-Generator	gen-[präfix]-site.js	gen-[präfix]-site.js