CLMS Product Filenaming - Design Principles

Copernicus Land Monitoring Service - Technical Library

Author: **European Environment Agency (EEA)**

Date: **2026-11-06**

Version: **1.2.0**

Table of contents

1. CLMS Product Filenaming - Design Principles 1
   - 1.1 Purpose and Scope 1
   - 1.2 Principle 1 - Filename Structure: fields, delimiter, extension 1
   - 1.3 Principle 2 - Allowed Characters: uppercase, digits, underscore, hyphen 1
   - 1.4 Principle 3 - Maximum Filename Length 2
   - 1.5 Principle 4 - Delimiter: Underscore `_` between fields, hyphen `-` within fields 2
   - 1.6 Principle 5 - Field Order: Variable hyphen-appended to the product code, token count invariant 2
   - 1.7 Principle 6 - Programme Prefix: `CLMS_` mandatory, no exceptions 3
   - 1.8 Principle 7 - Product Code: Short, uppercase, registered 3
   - 1.9 Principle 8 - Temporal Coverage: Format matches product type 3
   - 1.10 Principle 9 - Resolution: `R{value}[unit]` for raster, `V{value}[unit]` for vector and point grid ... 4
   - 1.11 Principle 10 - Spatial Extent: Identifies the area covered by this specific file 5
   - 1.12 Principle 11 - EPSG Code: Bare code, required for projected CRS, omit when CRS is implicit . 6
   - 1.13 Principle 12 - Version: `V{XX}-R{YY}` single hyphenated field 6
   - 1.14 Principle 13 - Variable: Hyphenated composite appended to the product code 6
   - 1.15 Principle 14 - Production Date: Metadata-only unless operationally required 7
   - 1.16 Principle 15 - Sensor Token: Omit unless multi-sensor discrimination is required 7
   - 1.17 Principle 16 - Quality Layers: Separate file, hierarchical `-QFLAG` suffix 7
   - 1.18 Principle 17 - Schema Registration: parsEO schema BEFORE CDSE onboarding 7
   - 1.19 What Cannot Be Standardised 7
   - 1.20 Principles Summary 8
2. Change Log 9

Contact:

European Environment Agency (EEA)
Kongens Nytorv 6
1050 Copenhagen K
Denmark
https://land.copernicus.eu/

1. CLMS Product Filenaming - Design Principles

Grounded in the parsEO schema registry, CDSE product catalogue, and current filenaming conventions.

1.1 Purpose and Scope

These principles govern the filenaming of **all new CLMS product deliveries** - new products, reprocessings, and new versions of existing products. They describe the target convention; existing files on CDSE under legacy names remain as-is but are non-compliant.

1.2 Principle 1 - Filename Structure: fields, delimiter, extension

Rule: A CLMS filename consists of a stem (base name) and if required (e.g. not folder names) an extension, separated by a period `.`:

```
{stem}.{extension}
```

The stem is composed of **fields** - descriptive elements separated by underscore `_`. Each field carries a specific meaning:

```
field1_field2_field3_…_fieldN.ext
```

Fields may not be empty. A field delimiter may not appear at the beginning or end of a filename, nor twice in a row.

Extension: The file extension (.tif, .gpkg, .xml) indicates the file format and is not part of the naming convention.

Compliant: `CLMS_VLCC-GRA_20210101P1Y_R10m_E27N48_3035_V01-R00.tif`

Non-compliant: `_CLMS_VLCC-GRA_…` (leading delimiter), `CLMS_VLCC-GRA__20210101P1Y_…` (double delimiter)

1.3 Principle 2 - Allowed Characters: uppercase, digits, underscore, hyphen

Rule: Filenames MUST use only the following ASCII characters:

Character	ASCII	Usage
`A–Z`	65–90	Uppercase letters only (no lowercase)
`0–9`	48–57	Digits
`_`	95	Field delimiter
`-`	45	Within-field separator

The period `.` (ASCII 46) is reserved exclusively for the file extension separator. Spaces, lowercase letters, and all other special characters are NOT permitted.

Rationale: A restricted character set enables deterministic, case-insensitive parsing across all platforms. Uppercase-only avoids case-sensitivity issues between Linux (case-sensitive) and Windows/macOS (case-insensitive).

Compliant: `CLMS`, `V01-R00`, `054-0154-IW1`, `R20-5m`, `VLCC-GRA`

Non-compliant: `clms_EGMS`, `V01_r00`, `P20x5m`, `data file.tif`

1.4 Principle 3 - Maximum Filename Length

Rule: The complete filename (stem + extension) MUST NOT exceed 255 characters. The fully qualified file name (including path) SHOULD NOT exceed 260 characters.

Recommendation: Keep filenames under 100 characters to safely accommodate both file name and storage path across all platforms.

Rationale: The 255-character limit is the smallest common denominator across Windows (260 char FQFN), macOS (1024), and Linux (4096). The 100-character recommendation preserves room for deep directory structures.

1.5 Principle 4 - Delimiter: Underscore _ between fields, hyphen - within fields

Rule: Use _ to separate all top-level fields. The hyphen - is the **within-field separator**: it joins the variable (and sub-variables) to the product code (`VLCC-GRA`, `WSI-SP-SCD`), joins version to revision (`V01-R00`), and appears where a notation itself requires it (e.g., `054-0154-IW1` for IW burst tile identifiers, `C2018-2021` for change periods). It is never a cross-field delimiter.

Rationale: Consistent underscore delimiters make tokenisation deterministic across all product families. The hyphen carries all intra-field structure, so splitting on _ always yields the same field count within a product family.

Compliant:

- `CLMS_VLCC-GRA_20210101P1Y_R10m_E27N48_3035_V01-R00.tif`
- `CLMS_EGMS-L2a-A_20190101P5Y_R20-5m_054-0154-IW1_V01-R00.tif` (hyphens inside `EGMS-L2a-A` and `054-0154-IW1` are semantic)
- `CLMS_CLCPLUS-LCU_20230101P1Y_R10m_E48N37_3035_V01-R00.tif`

Edge case: Hyphenated geographic names (e.g., “Bourg-en-Bresse”) is a single token; the hyphen is part of the name, not a delimiter.

1.6 Principle 5 - Field Order: Variable hyphen-appended to the product code, token count invariant

Rule: Fields appear in a fixed order per product family. The variable is **not a standalone field**: it is hyphen-appended to the product code in position 2 (`{CODE}-{VARIABLE}[-{SUBVARIABLE}]`). Underscore-separated token count is invariant across a product family - a sub-product identifier is never a separate positional field; it is encoded within the code-variable compound:

Canonical template:

```
CLMS_{CODE}-{VARIABLE}[-{SUBVARIABLE}]_{TEMPORAL}_{RES}_{EXTENT}_{EPSG}_{VERSION}-{REVISION}
```

Sub-products within a family (e.g., SP, SWS, WDS within WSI) extend the compound with a further hyphen (`WSI-SP-SCD`, `WSI-WDS-SSC`). This keeps the underscore token count identical for every file in the family.

Rationale: Code and variable together identify **what** the file is; the remaining fields identify **when**, **where**, and **which version**. Placing the full product identity at position 2 groups all files of a product and its layers together in directory listings, and a parser always knows which position holds which field - no branching logic based on which sub-product happens to be present.

Compliant:

- `CLMS_VPP2-ST-PPI_20240101P10D_R10m_T33UVS_V01-R00.tif`
- `CLMS_WSI-SP-SCD_20200901P1Y_R20m_T38TKL_V01-R00.tif`
- `CLMS_WSI-WDS-SSC_20210217T053159_R60m_T32TNS_S1B_V01-R00.tif` (sensing datetime, sub-product in the code-variable compound)

1.7 Principle 6 - Programme Prefix: `CLMS_` mandatory, no exceptions

Rule: Every CLMS product filename MUST begin with `CLMS_`. No exceptions. No product is exempt. This applies to all new deliveries, reprocessings, and new product versions across the entire CLMS portfolio (EEA, JRC, contractors).

Rationale: The prefix makes programme identity parseable from the filename alone. Any file without `CLMS_` is immediately identifiable as non-conformant.

Compliant:

- `CLMS_VLCC-GRA_20210101P1Y_R10m_E27N48_3035_V01-R00.tif`
- `CLMS_VPP2-ST-PPI_20240101P10D_R10m_T33UVS_V01-R00.tif`
- `CLMS_UA-LCU_20210101P1Y_V025ha_DK004L3_3035_V01-R00.gpkg`
- `CLMS_CLCPLUS-LCU_20180101P1Y_R10m_EUROPE_3035_V01-R00.tif`
- `CLMS_CLC-LCU_20180101P1Y_V25ha_EUROPE_3035_V01-R00.gpkg`
- `CLMS_EGMS-L3-U_20190101P5Y_R100m_E28N49_3035_V01-R00.tif`

1.8 Principle 7 - Product Code: Short, uppercase, registered

Rule: Product codes must be registered in the CLMS product code registry before schema authoring. Uppercase alphanumeric only. Prefer 4–7 characters. Retain the `HR` prefix only when it is part of the established acronym.

Compliant: `VLCC`, `VPP2`, `WSI`, `EUHYDRO`, `CLCPLUS`, `UA`

Non-compliant: `HRVPP2` (redundant HR), `VPP-2` (hyphen in code), any unregistered code

---

1.9 Principle 8 - Temporal Coverage: Format matches product type

Rule: Every product except a change product uses one of the first two forms below - the sensing instant or the period. No other temporal form is permitted. Change products use the third form.

Situation	Format	Examples	Definition
Sensing instant	`{YYYYMMDD}T{HHMMSS}`	`20210217T053159`	Single acquisition at a specific date+time (e.g. satellite scene). Zero-duration temporal instant. Uses bare ISO 8601 `dateTime` lexical form - the `T` separator and fixed-width format make it self-identifying, so no semantic prefix is needed.
Period	`{YYYYMMDD}P{duration}`	`20240101P10D`, `20240101P1M`, `20240101P1Y`, `20210101P1Y`, `20190101P5Y`	A file whose temporal coverage is the period that starts on the given date and lasts the given ISO 8601 duration. Carries composites, status years, multi-year series and forecasts alike. No prefix letter.
Change	`C{YYYY}-{YYYY}`	`C2018-2021`	A map of transitions or differences between two reference years. The two years are the start and end of the change period. The output is a 2D change layer (not a time-series stack).

Temporal reference standard: All temporal formats follow **ISO 8601-1:2019** (Date and time - Representations for information interchange). The `P{duration}` suffix uses the ISO 8601 duration format (`P10D`, `P1M`, `P1Y`). Reference: https://www.iso.org/standard/70907.html

Edge case: Multi-seasonal products encode the season in the code-variable compound (`-S1`, `-S2`), not in the temporal field. This is consistent with external standards (CF Conventions, NASA EOSDIS) which treat season as a data attribute, not a temporal primitive.

Compliant examples:

- `CLMS_VPP2-ST-PPI_20240101P10D_…` (period, dekadal)
- `CLMS_VLCC-GRA_20210101P1Y_…` (period, status year)
- `CLMS_UA-LCU_C2018-2021_…` (change between two reference years)
- `CLMS_WSI-WDS-SSC_20210217T053159_R60m_T32TNS_S1B_V01-R00.tif` (sensing datetime)
- `CLMS_EGMS-L3-U_20190101P5Y_R100m_E28N49_3035_V01-R00.tif` (period, multi-year series)

1.10 Principle 9 - Resolution: `R{value}[unit]` for raster, `V{value}[unit]` for vector and point grid

Rule: Raster and gridded products use `R{value}[unit]` with no zero-padding: `R10m`, `R1km`, `R100m`. The value is the integer pixel/grid spacing in metres. Irregular grid spacings join both values with a hyphen within the field: `R20-5m` (e.g., EGMS L2 20×5 m posting).

The `V` prefix covers two cases:

- **Vector minimum mapping units** - `V{value}ha` (e.g., `V025ha` for Urban Atlas 0.25 ha, `V25ha` for CLC 25 ha). Sub-1 ha values use a leading zero with no decimal point.

- **Point grid posting spacing** - `V{value}m` for regular grids (e.g., `V100m` for EGMS L3) or `V{value}-{value}m` for irregular grids (e.g., `V20-5m` for EGMS L2 point data)

Only `R` and `V` exist - there is no separate point-grid prefix.

Compliant: `R10m`, `R1km`, `R100m`, `R20-5m`, `V025ha`, `V25ha`, `V100m`, `V20-5m`

Edge case: Products with no meaningful spatial resolution (e.g., nation-level statistical vector layer) omit the resolution token.

1.11 Principle 10 - Spatial Extent: Identifies the area covered by this specific file

Rule: This field identifies the spatial extent of the individual file - not the overall product coverage. It can be a grid tile, a satellite scene, an administrative boundary, or a continental code:

Extent type	Description	Examples
MGRS	Sentinel-2 raster tile	`T33UVS`
LAEA 100km	Pan-European composite on INSPIRE/ETRS89-LAEA grid	`E27N48`
IW burst	Sentinel-1 swath (burst-level)	`054-0154-IW1`
FUA code	Functional Urban Area, city-level vector	`DK004L3`
Country code	Nation-level vector	`DE`, `IT`
Delivery Unit	Tiling system for large-area products	`DU001`
Continental	Full European continent coverage	`EUROPE`
Hemispherical	Global product subset for one hemisphere	`NORTHERNHEMISPHERE`, `SOUTHERNHEMISPHERE`
Global	Full global coverage	`GLOBAL`

Spatial extent identifiers are _-separated from adjacent tokens. Never fuse them with EPSG or resolution.

> **Note:** This field describes the extent of the **file**, not the product. A pan-European product may be delivered as individual MGRS tiles, each with a different spatial extent. Likewise, `EUROPE` means this specific file covers the entire continent - it does not mean the product is "pan-European" by design.

Compliant:

- `CLMS_VPP2-ST-PPI_…_T33UVS_…` (MGRS tile)
- `CLMS_VLCC-GRA_20210101P1Y_R10m_E27N48_…` (LAEA tile)
- `CLMS_CLCPLUS-LCU_20230101P1Y_R10m_E48N37_…` (LAEA tile)
- `CLMS_UA-LCU_20210101P1Y_V025ha_DK004L3_…` (FUA code)
- `CLMS_EGMS-L2a-A_…_054-0154-IW1_…` (IW burst)
- `CLMS_EUHYDRO-NET-LINE_EUROPE_…` (continental coverage)
- `CLMS_WSI-WDS-SSC_20210217T053159_R60m_T32TNS_S1B_V01-R00.tif` (MGRS, SAR sensor)

1.12 Principle 11 - EPSG Code: Bare code, required for projected CRS, omit when CRS is implicit

Rule: Standalone _-separated token containing the bare EPSG code with **no zero-padding** (`3035`). Required for products with a projected CRS (e.g., LAEA 100km extents). Omit when the CRS is implicit in the spatial extent system (e.g., MGRS tiles imply UTM, so EPSG is redundant). Also omit for CRS-independent files (metadata XML, archive containers).

Rationale: The bare code is the EPSG identifier exactly as registered (https://epsg.org) - `3035`, `4326`, `32633` - so the token matches the authority and needs no transformation for lookups. EPSG codes are 4–5 digits; the field is bounded, and the surrounding fixed field order keeps parsing deterministic without artificial padding.

Compliant:

- `CLMS_VLCC-GRA_20210101P1Y_R10m_E27N48_3035_V01-R00.tif` (LAEA tile, EPSG 3035 required)
- `CLMS_CLCPLUS-LCU_20230101P1Y_R10m_E48N37_3035_V01-R00.tif`
- `CLMS_VPP2-ST-PPI_20240101P10D_R10m_T33UVS_V01-R00.tif` (MGRS tile - EPSG omitted, CRS implicit)

Non-compliant: `03035` (zero-padded), `EPSG3035` (prefixed)

1.13 Principle 12 - Version: `V{XX}-R{YY}` single hyphenated field

Rule: A single compound token: `V{XX}` (major, 2-digit zero-padded), hyphen, `R{YY}` (revision, 2-digit zero-padded). Never use packed `V{XXX}`.

Compliant: `V01-R00`, `V01-R01`, `V02-R00`

1.14 Principle 13 - Variable: Hyphenated composite appended to the product code

Rule: The variable identifies the measured or mapped quantity. It is hyphen-appended to the product code at position 2, forming the product identity compound:

```
{CODE}-{PRODUCT_TYPE}[-{PARAMETER}][-{SEASON}][-QFLAG]
```

- `PRODUCT_TYPE`: layer type code (`ST`, `VPP`, `GPP`, `SP`, `LCU`)
- `PARAMETER`: specific measurement (`PPI`, `SOSD`, `TOTAL`, `SCD`)
- `SEASON`: `S1` or `S2` for multi-seasonal products
- `QFLAG`: literal suffix for quality flag layers - always last

`LCU` (Land Cover / Land Use) is the shared variable for all status and change land cover/use products (UA, CLC, CLCPLUS, RZ, ...); status vs change is expressed by the temporal field (`{YYYY}0101P1Y` vs `C{YYYY}-{YYYY}`), not by the variable.

Compliant:

```
VPP2-ST-PPI          → VPP2-ST-PPI-QFLAG
VPP2-VPP-SOSD-S1     → VPP2-VPP-SOSD-S1-QFLAG
VPP2-GPP-TOTAL       → VPP2-GPP-TOTAL-QFLAG
WSI-SP-SCD           → WSI-SP-SCD-QFLAG
UA-LCU / CLC-LCU / CLCPLUS-LCU
```

1.15 Principle 14 - Production Date: Metadata-only unless operationally required

Rule: Omit production date from filenames. Include it only when multiple production runs for the same temporal period coexist in the same distribution channel. When present: `{YYYYMMDD}` as the last token before the extension.

Compliant: `CLMS_EUHYDRO-NET-LINE_EUROPE_3035_V02-R00_20261015.gpkg`

1.16 Principle 15 - Sensor Token: Omit unless multi-sensor discrimination is required

Rule: No standalone sensor token in the filename. The sensor is implicit in the product code. Include a sensor token only when the product supports multiple sensor sources distinguishable at file level (e.g., SAR platform identity `_S1B_`).

Compliant:

- `CLMS_WSI-SP-SCD_20200901P1Y_R20m_T38TKL_V01-R00.tif`
- `CLMS_WSI-WDS-SSC_20210217T053159_R60m_T32TNS_S1B_V01-R00.tif` (S1B sensor token required for SAR discrimination)

1.17 Principle 16 - Quality Layers: Separate file, hierarchical `-QFLAG` suffix

Rule: Quality layers are separate files. Same stem as the data layer, with `-QFLAG` appended to the appropriate hierarchical level of the code-variable compound. No other field may differ between a data file and its quality file.

The `-QFLAG` suffix attaches at the level it applies to:

- `VPP2-VPP-SOSD-QFLAG` - quality applies to the VPP-SOSD product only
- `VPP2-VPP-QFLAG` - quality applies to all VPP sub-products (SOSD, TOTAL, SEASONAL, etc.)
- `VPP2-ST-PPI-QFLAG` - quality applies to the PPI measurement only

Compliant:

```
CLMS_VPP2-ST-PPI_20240101P10D_R10m_T33UVS_V01-R00.tif       ← data
CLMS_VPP2-ST-PPI-QFLAG_20240101P10D_R10m_T33UVS_V01-R00.tif ← quality
```

1.18 Principle 17 - Schema Registration: parsEO schema BEFORE CDSE onboarding

Rule: Every new CLMS product must have a registered product code, a parsEO schema with `status: current`, and a naming crosswalk before the first CDSE delivery. The schema defines the filename; CDSE filenames are not reverse-engineered into schemas after the fact.

1.19 What Cannot Be Standardised

Some aspects genuinely differ between product families and should not be forced into a single mould:

1. **Raster vs. vector/point-grid resolution** - `R10m` and `V25ha` are different physical quantities; `V` covers both vector MMU (`V{value}ha`) and point grids (`V{value}m`)
2. **Pan-European vs. local tile encoding** - LAEA grid tiles (`E27N48`) for pan-European products, FUA codes (`DK004L3`) for city-level vector - these are incompatible spatial encodings, both valid
3. **Annual status vs. scene-level temporal format** - `{YYYY}0101P1Y` for static maps, `{YYYYMMDD}T{HHMMSS}` for single acquisitions - the right format depends on the product's temporal nature
4. **Multi-resolution families** - the set of valid resolution values varies per product; each schema enumerates its own
5. **Production date inclusion** - reflects operational policy; included when corrections under the same version are expected

---

1.20 Principles Summary

#	Principle	Rule
1	Filename structure	Fields separated by _, stem.extension
2	Allowed characters	`A-Z`, `0-9`, _, `-` only; no lowercase, no spaces
3	Max filename length	≤255 chars stem+extension, recommend ≤100
4	Delimiter	_ between fields; `-` within fields (code-variable compound, version-revision, notations)
5	Field order	Variable hyphen-appended to product code at position 2; canonical template defines field sequence
6	Prefix	`CLMS_` mandatory for all products, no exceptions
7	Product code	Short, uppercase, registered in code registry
8	Temporal	`{YYYYMMDD}P{duration}` (period), `{YYYYMMDD}T{HHMMSS}` (instant), `C{YYYY}-{YYYY}` (change)
9	Resolution	`R{XX}m` (raster/grid), `V{value}ha` (vector MMU), `V{value}m` (point grid) - hyphen for irregular grids
10	Spatial extent	MGRS / LAEA / IW burst / FUA / country / DU / EUROPE - identifies the area of the specific file, not the product coverage
11	EPSG	Bare EPSG code, no zero-padding (`3035`); required for projected CRS, omit when CRS is implicit in the spatial extent
12	Version	`V{XX}-R{YY}` single hyphenated field; no packed `V{XXX}`
13	Variable	`{CODE}-{TYPE}[-{PARAM}][-{SEASON}][-QFLAG]` compound at position 2
14	Production date	Metadata-only unless operationally required
15	Sensor	Omit unless multi-sensor discrimination needed
16	Quality	Separate file, hierarchical `-QFLAG` suffix, identical stem
17	Registration	parsEO schema before CDSE onboarding

Document status: Draft for review. Grounded in the parsEO schema registry and VPP2 naming crosswalk.

2. Change Log

Date	Version	Summary
2026-11-10	1.2.0	• Updated document date in frontmatter. • Major revision of Principle 8 (Temporal Coverage) to consolidate 'Aggregated', 'Status', and 'Timeseries' formats into a single 'Period' format. • Removed `A`, `S`, and `T` prefixes from temporal fields, simplifying the rule set. • Updated all filename examples throughout the document to reflect the new temporal field convention. • Revised the summary table at the end to align with the new temporal field principles.
2026-10-29	1.1.0	Initial publication of the 'CLMS Product Filenaming - Design Principles' document, detailing rules for filename structure, allowed characters, delimiters, field order, and product-specific tokens.