

Workflow to get the Application Family via AI

Goal is an automated process which exports the necessary data and sends it to the AI, receives the csvs and triggers synchronization

Manual Process:

▲ Claude Excel Plugin can only work directly with the **one spreadsheet that's currently open**

Exports

1. Export all Application [AE] objects without application family into csv "Snow_Applications_and_Family"

a. AQL:

```
1 | "Family Name" IS EMPTY AND "Installable Application" IS EMPTY
```

2. Export all Installable Application objects as csv "Family_Master"

a. all objects

3. combine both csvs in 1 xlsx file with tab names "Snow_Applications_and_Family" (Application[AE]) and Family_Master (Installable Application) and 1 empty Tab "unique_names"

AI Tooling: @Stanislav Ivanov

- Microsoft Excel + Claude for Microsoft Excel plugin (requires min. Pro subscription for claude.ai)
- All data should come in one excel file i

AI (example) Prompt: @Stanislav Ivanov

Your start base - two input files (CMDB export + SNOW/AE catalog with ~598 canonical names)

```
1 # CMDB Application Name Cleanup - Full Pipeline Prompt
2
3 ## How It Works
4
5 The customer has two files:
6 1. CMDB Export ('Application_AE.csv') - ~6600 rows of raw installed software with 'Name', 'Manufacturer', and other metadata
7 2. SNOW/AE Application Catalog - a reference list of ~600 canonical "Application Family" names from ServiceNow / Asset Explorer
8
9 The goal: map every CMDB entry to a clean Application Family name, then produce a deduplicated list of ~2800 unique families.
10
11 ---
12
13 ## The Prompt
14
15 > Upload both files and paste everything below:
16
17 ---
18
19 I'm giving you two files:
20
21 1. CMDB Export CSV - each row is an installed software entry. Key columns: 'Name' (raw application name), 'Manufacturer' (vendor).
22 2. SNOW/AE Application Catalog - a reference list of canonical application family names. This is my authoritative normalization
    source.
23
24 Process the CMDB export through these 4 stages. Output the result of each stage as a column, plus the final deduplicated list.
25
26 STAGE 1 - Strip version/build noise → 'Application Family'
27
28 For each 'Name', remove trailing version, build, architecture, and language info to extract the core product identity.
29
30 Remove these patterns:
31 - Semantic versions: '7-Zip 24.09 (x64 edition)' → '7-Zip'
32 - Build numbers: 'ActivePerl 5.10.1 Build 1007' → 'ActivePerl'
33 - Architecture tags: 'Adobe Acrobat (64-bit)' → 'Adobe Acrobat'
34 - Year-as-version: 'Adobe After Effects 2025' → 'Adobe After Effects'
35 - Model + version combos: '400 Series Utility - FCS541 v5.6.0.4' → '400 Series Utility'
36 - Language suffixes: 'Adobe Acrobat Reader DC - Deutsch' → 'Adobe Acrobat Reader DC'
37 - Hostnames in brackets: 'Access Console 23.3.4 [hostname.com]' → 'Access Console'
38 - Leading/trailing special chars: '>pilot 4.0< 1.19.0.0 (0NA04811_WZV)' → 'pilot 4.0'
39
40 Keep intact words that ARE the product identity: 'Microsoft 365', '3D Viewer', 'IIS 10.0 Express'.
41
42 STAGE 2 - Match against SNOW/AE catalog → 'SNOWApplication Family' + 'Not Matched'
43
44 Take each 'Application Family' from Stage 1 and fuzzy-match it against the SNOW/AE catalog:
45
46 - Match found → set 'SNOWApplication Family' to the matched catalog entry
```

```

47 - **No match** → leave `SNOWApplication Family` empty, set `Not Matched` to the Stage 1 value
48
49 Use semantic/fuzzy matching – the Stage 1 value won't always exactly match the catalog. Examples:
50 - `Adobe Acrobat DC` should match SNOW's `Adobe Acrobat`
51 - `3D PDF Maker Smart v5` should match SNOW's `3D PDF Maker Smart`
52 - `Admin By Request Workstation` should match SNOW's `Admin By Request`
53
54 Set `Final Application Family` = SNOW value if matched, else the Stage 1 Application Family value.
55
56 ### STAGE 3 – Clean and normalize → `Cleaned Application Family`
57
58 Apply these rules to `Final Application Family`:
59
60 **3a. Manufacturer-level ecosystem collapsing:**
61 All products from `Adobe Systems` / `Adobe Inc` → **Adobe** (Acrobat, Photoshop, After Effects, Creative Cloud, etc. are one
licensing ecosystem).
62
63 **3b. OS grouping:**
64 All `Microsoft Windows` variants (10, 11, Server 2019, Server 2022, etc.) → **Windows**
65
66 **3c. Sub-product merging:**
67 - `3DEXPERIENCE Exchange`, `3DEXPERIENCE Launcher`, `3DEXPERIENCE Marketplace` → **3DEXPERIENCE**
68 - `ABBYY FineReader 12 Sprint` → **ABBYY FineReader**
69 - `ADManager Plus Free Tools` → **ADManager Plus**
70 - Remaining version fragments: `400 Series Utility - FCS541` → **400 Series Utility**
71
72 **3d. Localized variant merging** (use the English name):
73 - `3D ビューアー`, `3D-Viewer`, `3D Viewer` → **3D Viewer**
74 - `App-Installer`, `App Installer` → **App Installer**
75 - `Dokumente`, `Documentos`, `Documents`, `Docs` → **Docs**
76 - `Feuilles de calcul`, `Hojas de cálculo`, `Tabellen`, `Spreadsheet` → **Spreadsheet**
77
78 **3e. Renamed product normalization:**
79 - `Acronis True Image` → **Acronis Cyber Protect Home Office**
80
81 **3f. What to keep SEPARATE (do NOT collapse):**
82 - **Microsoft individual products:** `Microsoft Teams`, `Microsoft Excel`, `Microsoft Edge`, `Azure Data Studio`, `App Installer`, `AV1
Video Extension`, etc. – distinct products
83 - **Siemens product lines:** `SIMATIC`, `TIA`, `SINEMA RC`, `SIMARIS`, `LOGO!Soft Comfort`
84 - **Linux packages** (Red Hat, Ubuntu, CentOS): keep each package as its own family (`libcurl`, `systemd`, `bash`, etc.)
85 - **Different products from same manufacturer:** `aconso Client` vs `aconso OfficeConnector & TransferService` – separate
86
87 ### STAGE 4 – Deduplicate and output
88
89 1. Collect all `Cleaned Application Family` values
90 2. Remove exact duplicates
91 3. Sort alphabetically (case-insensitive)
92 4. Output as CSV with header `Unique Application Name`
93
94 ### OUTPUT FORMAT
95
96 Produce **two outputs**:
97
98 **Output 1 – Full mapping table** (same row count as input CSV):
99
100 | Name | Manufacturer | Application Family | SNOWApplication Family | Not Matched | Final Application Family | Cleaned Application
Family |
101 |-----|-----|-----|-----|-----|-----|-----|
102
103 **Output 2 – Deduplicated list** (single-column CSV):
104 ...
105
106 Unique Application Name
107 7-Zip
108 @salesforce/cli
109 ABBYY FineReader
110 ...
111 ...
112
113 ### RULES
114 - Process ALL rows. Do not skip, sample, or truncate.
115 - Every output name must be derivable from at least one input `Name`.
116 - When in doubt whether to merge → keep separate.
117 - Use `Manufacturer` as a grouping signal to confirm family groupings.
118 - Expected: ~6600 input rows → ~2800 unique families. If you get <2000 you're over-collapsing; >3500 you're under-normalizing.

```

- Based on the output - please DO verify (human) - further strengthen the prompt with iterating until you're satisfied.
- Final list becomes the **source of truth**, which then can be used for **automated process**.

Imports:

Sync the new csv with the Application Families:

Import: AI Application Family, exchange the import file

Sync the Application [AE]:

Import: AI Application, exchange the import file

Automated Process

To be discussed with the AI Chapter (Dominic and Stani)



