



链滴

规范性文件、文件夹命名系统，用以解决文件整理、存档、检索的问题【元数据文件命名法】

作者：[HaujetZhao](#)

原文链接：<https://ld246.com/article/1635066725430>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

背景

我曾听过一句有哲理的话，具体找不到了，它的大意是：

没有良好命名的文件，基本就是没有意义的文件。

但在实际生活中，给需要存档、方便以后寻找的文件起名、整理，也是一件让我头疼的事。每次取名都要对名字思考：

- 这个名字恰当吗？
- 方便我以后检索到吗？
- 名字足够有概括性以至于以后一眼就看懂吗？
-

真是难呐！

目前我在网上能找到的命名法，都是类似这样的：

命名结构：**项目名词（或编号）+文件名词+文件作者+日期+版本号.ext**

例如：2016年公司部门工作总结_营销部_鹏_20170101_V1.0.doc

文件名称由五部分组成：

- 第一部分为阐述文件主题，观其名知大意；
- 第二部分为文件所属类别，如在单位工作的写工作部分、学生人群可写班级等；
- 第三部分为文件创建者；
- 第四部分为当前文件的日期；
- 第五部分为文件阶段标识，用于版本管理。

但我觉得不好用！或者说，不够用！现实的文件情况可比这复杂多了！

在前 2021 年 10 月 23 日，我提出了 [规范性图片文件名整理系统的构思](#)，用以解决图片整理、检索问题，我认为这个方法很好，非常周到，只是它局限在了管理图片、视频方面，如果扩展一下，可以用到大部分文件、文件夹的命名管理当中，因此，经过思索，我对其进行扩展，得到一套详细、好用好理解的命名法。

一个方法，为了方便交流，最好还是取个名字，因为网上还没人起过这样的名字，我就先大言不惭地这样一个名：**元数据文件命名法**

它是一个为文件、文件夹命名的规则，覆盖了以下适用范围：

- 照片、视频、设计素材
- 工作项目
- 书籍、文档
- 软件安装包
- 录像素材
- 科研数据文件

- (几乎一切需要存档的文件)

不适用的地方：编程源文件的命名

命名原则

元数据文件命名法 提倡，在文件、文件的命名中，可选地、有序地包含以下以下 7 个 **元数据块**：

1. 时间：年、月、日、时分秒，根据实际情况决定精度
2. 前缀：可以是数字序号、大的分类；不常用，主要目的是为文件进行手动排序
3. 标题：文件的主标题，描述文件内容的本质
4. 版本：版本号，对于需要迭代的项目文件很重要
5. 标签：标题之外的补充信息，例如地点、文件性质
6. 人物：可以是人名、组织名、宠物名、作者、来源答主
7. 备注：评论性文字，杂谈、心得

每个元数据块之间，应当使用规范的、高可读性的 **块间分隔符**

在命名时，要注意文件系统对文件名的限制：

- NTFS 上最多 255 个字符
- 有些系统上最多 255 字节 (每个中文2字节)
- 不能使用的字符： < > / \ | : " * ?

宽松命名和严格命名

对于普通文档管理，文件名受到的约束很小，中文、空格、!@#\$% 等特殊符号可以使用。

但实际现实中，还有一些特殊场景，如：

- 科学计算
- 实验数据处理
- 剪辑素材

在这些方面，对文件命名的限制会更高，因为有些古老的工具，对于文件名中的特殊字符、中文字符空格，不能正确地处理，会造成错误。

为避免不兼容，最安全保险的文件命名又受以下限制：

- 不能有中文
- 主体为数字、英文字母 (ASCII 符号)
- 可输入英文符号，但是避免： : ~ ! @ # \$ % ^ & * () ` ; < > ? , [] { } ' " ' "
- 可使用的安全符号： - _ .

因此，**元数据文件命名法** 分为 **宽松** 和 **严格** 两小类。它们遵循相同的原则，只是 **严格** 版在字符使用更加安全，更适合科学研究领域。

元数据文件命名法【宽松】

命名示例

直接列举规则细节，会不方便理解，所以先以几个规范的 **示例** 帮助学习和理解：

(20181005-170201)婚纱照#苏州#太湖#夕阳@伊女士@胡先生@摄影李.jpg

(20190207-105303)合影#过年#走亲戚#老家@唐语&二姨家来走亲戚，和表姐合影.jpg
(20200102-130843)酥肉汤#美食#老家@李明@刘大龙&我和李明在他家，第1次学了炸酥肉，煮了
&他表弟在旁边捣蛋.mp4

(2020-03)销售数据#北京#广告业务@林佳一&她是本月的销冠.xlsx
(2020-03)销售数据#北京#服务器@黄元&这个月不太景气，有些下滑.xlsx

(2021-08-23)我还年轻，我还年轻#御音萌妹#翻唱@咩咩爱睡懒觉.mp4
(2021-05-05)【囍】嫁鸡随鸡，嫁狗随狗#舞蹈@-小D-biu.mp4

(2020-03)面试简历#后端@高净&小伙子技术相当可以，老板要定了.doc
(2020-03)面试简历#运维@林嵘宏&可能申请的岗位不大合适.doc

(2012)中国图书馆分类法简本#分类@国家图书馆编辑委员会.pdf
(2014)Talk Like TED#演讲@Carmine Gallo.epub
(2020)费曼学习方法#学习@Farnam Street Media.pdf

(2012)Adobe Photoshop CS6(v13.0.1.1)#便携版&从www.xxxx.com下载的.7z

(2015-09)裁员计划#已批准@人事部

基地扩建项目#todo@老张=规模有些大，得多开几次会
健身视频#routine#健康#运动
(2014)同学照片#人物@初中同学

(2021-05-13)004-玩具车(v1.1.1)#done#rhino@客户32&这个要求是真简单，就喜欢这样的客户
(2021-05-16)002-海贼王(v3.4.5)#todo#3ds Max@客户38&反稿了好多次，要求贼高，但客户又差钱

001-颜文字皮肤(v1.0)#键盘
001-颜文字皮肤(v1.6)#键盘&啊!!! 心态崩了，重构重构!
001-颜文字皮肤(v2.0)#键盘
001-颜文字皮肤(v2.1)#键盘#done&Nice，这是最终版了

宽松命名细则

提倡将以下项目可选地、按顺序地写入文件名

1. 时间（可选）

样式有：

(YYYYMMDD-HHMMSS)
(YYYY-MM-DD)

(YYYY-MM)
(YYYY)

例如：

(20210512-091500)
(2021-05-12)
(2021-05)
(2021)

如此设计的原因：

- 时间开头，方便按时间排序
- 时间放在括号中间，与后面文字部分形成明显界限，方便批处理获取时间信息
- 时间放在括号中美观呐
- **年月日** 之间使用 - 分隔，高可读性，用户看着不累
- 加上 **时分秒** 后，只在 **日期** 和 **时间** 之间用 - 分隔：
 - 整个时间戳有 14 位数字，便于用户区分 **日期** 和 **时间** 部分
 - 不能太长了，不然在资源管理器中显示效果不好

2. 前缀（可选）

可以是数字序号、大的分类，使用 - 分隔，例如：

01-
005-
022-
122-
属性1-
属性1-属性2-
属性1-属性2-属性3-

如此设计的原因：

- 用户有时会有 **手动排序** 的需求，通过在标题前加上数字序号、或者对文件特别重要的属性，可达到手工排序归类的效果
- 为了高可读性，前缀和标题间，应当有一个合法、但不常用的符号用作分隔，在中文输入法状态下，- 也是一个方便输入的字符

3. 标题（可选）

文字标题，用以表示该文件的最本质属性，不需要太多废话，多余的补充信息放到标签、评论中就可以

4. 版本（可选）

用以解决一个项目时常有多个版本的情况，样式是：

(v大版本号[.中版本号[.小版本号]])

例如：

(v1.1.1)
(v3.4.5)
(v1.6)
(v1)
(v6)

如此设计的原因：

- 版本号外加上 **括号**，显眼，好区分，耐看，与文件名中其它部分形成明显分隔
- 版本号中的 **v**，让人一眼就知道这几个数字表示的是版本
- 版本号一定要排在标题后，这样相同项目的不同版本，才能排列在一起
- 版本号一定要排在 **标签、评论** 前，这样才能让同项目的不同版本有序排列

5. 标签 (可选)

可以对文件性质、分类、地点进行补充，样式是：

#标签文字

例如：

#todo
#done
#后端
#已批准
#搁置
#学习方法
#北京
#New York

标签可以有多个，标签文字中可以包含空格

如此设计的原因是：在微博、空间、Twitter 等社交媒体上，# 已经被广泛地用作话题标签，在 Obsidian 等笔记软件中，也被用作标签的标识符，用户可以一眼就下意识地得知：这是一个标签。

6. 人物 (可选)

即名字，但不止于人名，可以是组织名、公司名、作者名、设计师名、部门名、博主名、答主名、摄影师名、客户名.....，样式是：

@名字

例如：

@人事部
@伊女士
@摄影李
@阿豪的阁楼
@John Legend

名字支持空格

如此设计的原因：

- 在社交网站上 @用户名 使用非常广泛，用户一看，就知道后面跟着的是个名字
- 文件是服务于人的，是人创造的，大部分文件总会对应到生活中的一个活的对象，它可以是人，也可以是组织。总之，在文件中加上名字，可以方便用户根据人名、组织名快速找到相关文件。
- 人名在文件名中的权重不用太高，主要是检索时用，应当排在后面些

7. 评论、备注（可选）

它的样式是：

&评论文字&第二行评论文字

其中，规定以 & 作为评论起始的分隔符，同时之后的 & 充当换行符

评论要尽可能简要，避免让文件名超过 255 个字符的限制

如此设计的原因：

- 用户有对文件进行补充性说明的需求
- 补充性文字的权重最低，应排在最后面
- 文件名长度有限，从这方面讲，也应当将可调整性最强的评论部分放到最后面
- 使用 = 将评论与前面的部分分隔开，方便用户、批处理程序区分
- 评论只有一坨不符合直觉，需要一个合法、不常用的词充当换行符

上述所有 7 个内容块的标识符号，使用的是：() - @ # &，除了 () 外，其余所有的符号都可以在中输入法下方便地输入

检索

经过这样的命名，基本上，没有什么文件是你找不到的了，例如我想要找所有 -小D-biu 在 2020 年视频，只需要在 Everything 或者 Listary 中输入：2020 @-小D-biu .mp4, Nice!

另外，上述的规范还特别方便批处理，这也是在设计这套命名时，非常注重的一个点

元数据文件命名法【严格】

与宽松版相比，原则不变，文件命名中应当可选地、按顺序地包含以下 7 个 **元数据块**：

1. 时间
2. 前缀
3. 标题
4. 版本
5. 标签
6. 人物

7. 备注

标题规范

科研标题

对于科研文件，提倡标题部分使用缩写，且应包含：

- 项目、实验名称
- 位置、空间坐标、样本序号
- 研究条件、样品处理方式
-

对不同的项目，标题的缩写应制定具体、合理的规范

录像素材标题

对于要用于剪辑的录像素材，提倡标题部分使用缩写，且应包含：

- 项目、主题名称
- 场景号
- 摄影机编号
- 素材尺寸、帧率
- 编码类型
-

对不同的项目，标题的缩写应制定具体、合理的规范

严格命名细则

在基于 [宽松命名细则](#) 的基础上，附加以下细则：

1. 仅使用数字、英文字符
2. 符号仅使用 `-_.`
3. 不能使用空格
4. 每个 **元数据块** 之间，应当使用 **块间分隔符**，推荐 `_` 或 `-`，例如：
 - `YYYYMMDD_Title_Author.ext`
 - `YYYYMMDD_Title_Author.ext`
 - `Title_Author.ext`
5. 每个 **元数据块** 内部，应当使用 **块内分隔符**，且应与 **块间分隔符** 不同，推荐 `.` 或 `-`，例如：
 - `YYYYMMDD_Attr1.Attr2.Attr3_Tag1.Tag2_Author1.Author2.ext`
 - `YYYYMMDD_Attr1-Attr2-Attr3_Tag1-Tag2_Author1-Author2.ext`

6. 年月日、时分秒之间，应当使用 **块内分隔符**，例如：

- YYYYMMDD-HHMMSS__Title__Author.ext

7. 连续使用多个单词时，推荐使用 **驼峰法** 或 **块内分隔符**（一定不要用空格），如：

- LongAndNecessaryTitle__Author.ext
- Long-and-necessary-title__Author.ext
- Long.and.necessary.title__Author.ext

科研文件命名示例

参考模板：

YYYYMMDD_Attr1.Attr2.Attr3.Attr4_Tag1.Tag2_Author1_Comment.ext
YYYYMMDD-HHMMSS__Attr1.Attr2.Attr3.Attr4__Tag1.Tag2__Author1.Autor2__Comment.ext

YYYYMMDD_Place.SampleNum.Procedure.Number_Tag1.Tag2_Author.ext
YYYYMMDD-HHMMSS__Place.SampleNum.Procedure.Number__Tag1.Tag2__Author.ext

具体示例：

20140623_FR3S.129C.2653_NewProgress_BD.JPG

日期：20140623，2014年06月23日

标题：FR3S.129C.2653.W

标题缩写释义：

FR3S：研究地点 FR3；Shallow，潜水域（S=Shallow, M=Middle, D=Deep）

129C：区域129，覆盖处理（C=Covered, U=Uncovered）

2653：照片序号

标签：NewProgress 有重要新进展，有别于其它普通数据

研究人员：BD（Bruce D）

录像素材命名示例

参考模板：

YYYYMMDD_Theme.Scene.Camera.ResFps.Codec.FileNum_Tags_Authors.ext
YYYYMMDD-HHMMSS__Theme.Scene.Camera.ResFps.Codec.FileNum__Tags__Authors__Comment.ext

标题缩写释义：

Theme：主题、项目

Scene：场景

Camera：录像设备

ResFps：分辨率、帧率

Codec：编码格式

FileNum：文件序号

示例：

20211025-102501_Proj1.S1.Rn4D.4k60.sLog3_BTS_Li.mov

20211025-102501_Proj1.S1.Rn4D.4k60.sLog3_BTS_Li.mov

时间: 2021年10月25日 10点25分01秒

标题: Proj1.S1.Rn4D.4k60.sLog3

标题缩写释义:

Prj1: 项目1 (Project 1)

S1: 场景1 (Scene 1)

Rn4D: 设备为如影4D

4k60: 分辨率4K, 帧率60

sLog3: Log类型

标签: BTS, Behind The Scene, 幕后

摄影师: Li

结语

要辩证

构思这样一个宽适用领域、高度弹性的命名规则是真费脑筋。

这样的命名法, 没有成为强制使用的初衷, 只是提供一个可供参考的模板, 在实际命名中, 要实际情实际分析, 灵活变更。渴望一条规则面对所有情况, 是不大可能的。

假设你在投简历的时候, 就不适用于这样的命名法, 而应当按公司要求的格式来, 例如:

- 应聘XX岗位-姓名-学校-手机号
- 姓名+学历+学校+实习时间+应聘岗位+意向城市+手机号

所以, 要辩证地看待该命名法, 具体情况具体分析!

如何用

有了这样一套方案, 不是说就一定要立即用到所有已存档的文件上, 它的意义是, 当你再需要给一系文件命名时, 会有一个科学合理的依据。

长文件名通常意味着更多的工作量, 目前的文件浏览器, 在重命名的操作上, 并不是很好用, 需要有些辅助性程序, 帮助规范命名, 可以是:

- 各种语言写的 GUI 程序
- Quicker 动作
- 批处理脚本

东西不复杂, 不用太多的技术, 但是打磨到好用的程度, 还是需要时间的。

参考

- [Best practices for file naming | Stanford Libraries](#)
- [Case study: File naming done well | Stanford Libraries](#)
- [文件的命名与组织](#)
- [Life Hack + 文件命名: 意义、要素与工具](#)