

T/JSCS

江苏省计算机学会团体标准

T/JSCS XXXX—XXXX

电力行业信息技术应用创新产品 应用开发技术规范

Technical Specification for Adaptation and Development of Information Technology

Application Innovation Products in the Power Industry

（征求意见稿）

2026-XXXX 发布

2026-XXXX 实施

江苏省计算机学会 发布

目 录

前 言 II

电力行业信息技术应用创新产品应用开发技术规范 1

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 2

5 编码要求 2

6 存储位置要求 3

7 日志要求 4

8 安全管理要求 4

9 软件包要求 4

10 桌面应用配置要求 7

11 单进程要求 8

12 多进程要求 8

13 兼容性要求 8

附录 A （资料性） 变更日志文件格式 10

附录 B （资料性） desktop 文件格式..... 11

附录 C （资料性） 符合性调研表 12

前 言

本文件按照GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利，本文件的发布机构不承担识别专利的责任。

本文件由江苏省软件和信息技术服务标准化技术委员会、江苏省计算机学会提出并归口。

本文件起草单位：

本文件主要起草人：

本文件为首次发布。

电力行业信息技术应用创新产品应用开发技术规范

1 范围

本文件面向电网行业在国产操作系统环境下进行的第三方应用开发活动，针对应用开发的编码、存储、日志、安全管理、软件包、桌面应用配置、单进程、多进程、兼容性等提出具体要求，旨在提升应用软件在电网信息系统中的兼容性、安全性与可维护性。

本文件适用于基于国产操作系统的调度控制系统、营销系统、生产管理系统、运维监控系统、综合办公系统等典型业务场景的应用开发，也为第三方应用程序的调试验证提供指导。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1 驼峰命名法 camel case

命名原则为第一个单词以小写字母开始，从第二个单词开始以后的每个单词的首字母均采用大写字母。

3.2 匈牙利命名法 Hungarian notation

命名原则是“变量名=属性+类型+对象描述”。

3.3 帕斯卡命名法 Pascal case

命名原则是由两个或两个以上单词连结在一起，每个单词首字母大写，单字之间不以空格断开或连字符（-）、下划线（_）连接。

3.4 下划线命名法 underscore case

命名原则是由两个或两个以上单词连接在一起，每个单词用下划线隔开且全部小写。

3.5 动态库 dynamic link library

一种不可执行的二进制程序文件，允许程序共享执行特殊任务所必需的代码和其他资源。

3.6 环境变量 environment variables

在操作系统中用来指定运行环境参数（如临时文件夹位置、系统文件夹位置等）的变量。

3.7 系统日志 system log

记录系统中硬件、软件和系统问题信息的日志，可用于监视系统事件。

3.8 软件包 software package

具有特定功能，用于完成特定任务的一个或一组程序。

3.9 变更日志 change log

对项目所作更改的详细记录，通常包括修复内容和新功能。

4 缩略语

下列缩略语适用于本文件：

PNG：可移植网络图形（Portable Network Graphics）

SVG：可缩放矢量图形（Scalable Vector Graphics）

UTF：通用字符集/编码转换格式（Unicode Transformation Format）

5 编码要求

5.1 编码语言要求

应用软件开发应使用符合 GB/T15272-1994 要求的 C 语言，宜使用 C++17、Java11、Python3.8 及以上等主流、安全语言和版本，不得使用已淘汰或存在安全隐患的语言或版本（如 Perl5.10 及更早版本）。

5.2 函数命名要求

在同一项目中应统一采用驼峰、匈牙利、帕斯卡或下划线命名法中的一种，命名规范已经选定，应形成项目命名约定文档，纳入版本管理，不得混用。

5.3 变量命名要求

变量命名还应遵守以下规则：

- a) 在单独代码行中声明每个变量；
- b) 避免无意义名称；
- c) 单字符变量仅在临时变量或循环计数时使用；
- d) 按需定义变量；
- e) 变量名称中的每个连续单词首字母大写；
- f) 避免缩写；
- g) 类名首字母，公有类以 Q 开头，公有函数以 q 开头；
- h) 含首字母缩写的变量仍采用驼峰命名法。

5.4 国际化要求

编码应遵守以下国际化规则：

- a) 应全面支持 UTF-8 和 BIG5 编码；
- b) 不得在源码中硬编码中文字符；

- c) 宜使用 `lupdate` 生成 `.ts` 文件，使用 `Linguist` 翻译生成 `.qm` 文件，通过加载 `.qm` 实现多语言；
- d) 文本资源应集中存放于资源文件（如 `.ts`、`.properties`）

5.5 变更日志要求

- a) 每个软件版本应对应一条变更日志；
- b) 格式应符合本标准附录 A 的变更日志格式要求。

6 存储位置要求

6.1 可执行文件存储位置

- a) 系统可执行文件：`/bin`；
- b) 系统软件可执行文件：`/usr/bin`；
- c) 用户可执行文件：`/usr/local/bin`；
- d) 应用软件可执行文件：`/opt/应用名/bin`。

6.2 动态库存储位置

- a) 系统软件动态库：`/usr/lib`；
- b) 应用软件动态库：`/opt/应用名/lib`，可通过符号链接方式部署；
 - 1) 编译时将 `/opt/应用名/lib` 加入 `rpath`、`runpath`；
 - 2) 应用软件运行不应依赖临时设置 `LD_LIBRARY_PATH`；
- c) `/lib` 和 `/usr/lib` 为默认搜索路径，其他路径通过：
 - 1) 在 `/etc/ld.so.conf` 添加后执行 `ldconfig`；
 - 2) 设置 `LD_LIBRARY_PATH`。

6.3 配置文件存储位置

- a) 全局配置：`/etc`，多文件时置于 `/etc/应用名/`；
- b) 用户配置：`/home/用户名/.config/应用名/`；
- c) 不得直接修改系统配置，应新建子配置；
 - 1) 环境变量脚本→`/etc/profile.d/应用名.sh`；
 - 2) `bash` 补全脚本→`/etc/bash_completion.d/应用名`；
 - 3) 新建用户模板→`/etc/skel/`；
 - 4) `AppArmor` 配置→`/etc/apparmor.d/`；
 - 5) `cron` 定时→`/etc/cron.d/应用名[-功能名]`；
 - 6) 新增配置变更日志→`/var/log/应用`

6.4 数据文件存储位置

- a) 系统软件数据：`/var/lib/软件名`；

- b) 应用软件数据: /opt/应用名/data/ (可自定义子目录)。

6.5 缓存文件存储位置

- a) 系统软件缓存: /var/cache/软件名/;
- b) 应用软件安装于/opt/应用名→缓存: /opt/应用名/cache/;
- c) 安装于/usr/share/应用名/→缓存: /var/cache/应用名/。

7 日志要求

7.1 系统日志

系统日志应符合以下要求:

- a) 日志输出不得影响系统运行;
- b) 不得泄露敏感信息;
- c) 应满足安全审计、备份、加密要求;
- d) 可用于定位问题。

7.2 应用日志

- a) 循环体内不打印 INFO;
- b) 日志打印应捕获异常, 防止程序退出;
- c) 宜使用占位符格式化, 避免字符串拼接;
- d) 日志文件至少保存 15 天;
- e) 应统一日志级别: DEBUG/INFO/WARN/ERROR/FATAL;
- f) 宜采用 JSON 结构化格式;
- g) 日志默认路径: /var/log/应用名/。

8 安全管理要求

- a) 含前端交互逻辑的可执行程序应以普通用户权限运行, 不得使用 sudo/pkexec 提权;
- b) 需特权操作时应采用前后端分离: 前端 (普通权限) 通过 dbus 等与后端 (管理员权限) 通信;
- c) 前端调用后端服务应进行身份鉴别和权限检查;
- d) 不得明文存储任何凭据, 应使用加密存储或密钥管理;
- e) 前后端通信应使用 TLS;
- f) 宜提供 SELinux/AppArmor 策略模板, 以降低最小权限;
- g) 敏感行为应记录审计日志, 便于事后追溯。

9 软件包要求

9.1 软件打包要求

- a) 按规则生成安装包，安装后可直接运行。
- b) 依赖声明完整，无需手工安装额外依赖；
- c) 包名格式：软件名_版本号_CPU 架构.包格式；
- d) 覆盖安装不得异常；
- e) 卸载时清理配置、数据、缓存；
- f) 安装包必须使用 GPG 或 SM2 签名，并提供公钥；
- g) 安装包必须提供 SHA-256 校验码；
- h) 安装脚本应检测最低系统资源（CPU、内存、磁盘）。

9.2 deb 包具体要求

9.2.1 版本号要求

版本号的唯一性应适用于源码包和二进制包，在任何源码包下，二进制包不能重复使用。

9.2.2 控制文件

包管理系统将控制数据存储在控制文件中，控制文件应使用 UTF-8 编码，由一个或多个字段段落组成，每个段落中包含多个数据字段，每个字段由字段名、紧跟着的冒号和该字段的数据/值组成。

一个段落不应出现一个字段有多个实例的情况。

9.2.3 软件包描述

每个 deb 包的 control 文件中包括 Description 字段，该字段应包含软件包的概要和扩展描述，还应描述关键依赖和其他软件包的冲突信息。

Description 字段不应包含配置或使用软件包的说明（即安装脚本、手册页、信息文件等），以及版权说明。

9.2.4 包变更日志

每个源码包中均应包含变更日志文件，包变更日志文件使用 UTF-8 编码，deb 包的变更应在变更日志文件中作出简短说明，包括新的软件包对于上游软件包作出的修改、其他变更以及软件包的更新信息，变更日志文件格式应符合附录 A 的规定。

9.2.5 编译脚本

编译脚本是一个可执行的 makefile 文件，应以#!/usr/bin/make -f 开头，便于使用名称调用而不是直接调用 make。

9.2.6 重复构建要求

为了实现重复构建，源码包的构建应符合以下要求：

- a) 在指定路径解压源码包；
- b) 已安装的构建依赖项版本；
- c) 指定的环境参数值；

- d) 确定构建架构;
- e) 确定主机的 CPU 指令架构。

9.2.7 软件包维护者

除了孤立软件包外，每个软件包都应有维护者，软件包的维护者应符合以下要求：

- a) 维护者信息应在控制文件的 **Maintainer** 字段中给出，包括名称和邮件地址；
- b) 如果一个人或者一个团队维护几个软件包，则维护者的名称和邮箱应在这些软件包的 **Maintainer** 字段中统一格式；
- c) 如果一个软件包的维护者是一个团队，并且使用一个共享的邮箱，则控制文件中的 **Uploader** 字段应指定至少一个人的私人邮箱地址；
- d) 如果软件包不再使用或者不可维护，则应将软件包删除。

9.2.8 维护者脚本

维护者脚本文件应以 **#!** 符号开头，这些脚本应是可读的、可被任何人执行的且不能是全局可写的。

维护者脚本调用的程序不应在其前面添加路径，这些程序的路径应在 **PATH** 环境变量中。

维护者脚本执行过程中，应符合以下要求：

- a) 幂等性要求，一次或多次请求调用，确保不会因为崩溃而导致调用没有完成；
- b) 如果维护者脚本与控制终端没有一起运行，则维护者脚本应回退到无交互行为状态；
- c) 维护者脚本运行成功应返回 0，运行失败应返回一个非零值；
- d) 安装、升级和移除到新版本软件包中的脚本名称以 **new-** 开头，升级和移除的旧版本软件包中的脚本名称以 **old-** 开头。

9.3 RPM 包具体要求

9.3.1 版本号要求

版本号的唯一性应适用于源码包和二进制包，在任何源码包下，二进制包不能重复使用。

命名应符合 **Name-Version-Release.{arch 类型}.rpm** 规范，**Name** 为软件包名称，**Version** 为软件版本号，**Release** 为发布序列号。版本号一般有如下要求：

- a) 仅当软件包比以前有较大改动时才增加 **Version** 号；
- b) 软件开发者对软件包打补丁时应将 **Release** 号加 1；
- c) 不要在 **Source** 语句中包含任何路径。

9.3.2 Spec 文件

Spec 文件应满足如下要求：

- a) 使用 **UTF-8** 编码，各段落之间应空一行；
- b) 文档开头以 **Summary** 开始，说明软件包基本信息等内容；
- c) 每个不同段落之间都以 **%** 作为开头；

d) 包括 Summary、Name、Version、Release、Vendor、Group、Source 等必要信息。

9.3.3 包变更日志

源码包的变更日志应加以说明，包括日期、维护者姓名、维护者联系方式、修改的具体内容等。

变更日志文件的格式应符合以下要求：

- a) 星期、月份均用英文形式的前 3 个字母，不应使用中文；
- b) 每条变更日志之间应有一空行。

变更日志文件格式应符合附录 A 的规定。

9.3.4 源码包构建原则

源码包构建应遵循以下规则：

- a) 只要存在 Spec 文件，开发者应在系统现有 Spec 文件上修改更新；
- b) 任何人无权删除其他人的变更日志文件和原始打包者的信息，但可追加自己的信息；
- c) Spec 文件应启用系统宏定义，保持一致性；
- d) 任何人都不应直接提供修改后的源代码，而应以补丁形式发布修改，在 Spec 里完成打补丁操作，一个补丁只解决一个问题，任何补丁的授权方式应和被修改源代码保持一致。

10 桌面应用配置要求

10.1 desktop 文件

桌面应用开发应使用 desktop 文件，用于在开始菜单、办公桌面等启动应用，desktop 文件应符合以下要求：

- a) 应以应用安装后软件包管理工具可查询的软件名为 desktop 文件名，以 .desktop 为文件后缀；
- b) 应符合 Linux 启动器 desktop 文件编写格式，desktop 文件格式示例见附录 B；
- c) 应用安装时，应将 desktop 文件放在“/usr/share/applications”目录下。

10.2 icon 文件

桌面应用开发应使用 icon 文件用于在开始菜单、任务栏等位置显示所需图标，icon 文件应符合以下要求：

- a) icon 文件名应为应用标识，图标文件格式为 PNG/SVG，其中 PNG 图标大小要求 128×128px 及以上或全尺寸；
- b) icon 若为 PNG 格式，在应用安装时应将 icon 文件放在“/usr/share/icons/hicolor”对应尺寸的目录下；
- c) icon 若为 SVG 格式，在应用安装时应将 icon 文件放在“/usr/share/icons/hicolor/scalable/mimetypes/”目录下。

10.3 系统及应用服务要求

系统及应用服务应符合以下要求：

- a) 系统级服务应提供 `systemd` 的服务配置文件，并将其放在“`/lib/systemd/system/`”目录下；
- b) 用户级服务应提供 `systemd` 的服务配置文件，并将其放在“`/usr/lib/systemd/user/`”目录下。

10.4 无障碍及用户体验要求

- a) 桌面应用应支持高对比度主题、字体缩放；
- b) 应提供可自定义快捷键；
- c) 应支持界面语言在线切换，配置文件建议存放

11 单进程要求

11.1 单用户单进程

单用户单进程情况下，应用宜使用文件锁的形式实现单进程，文件锁放到“`/run/user/用户 ID/`”目录下，并使用“应用名.pid”方式命名，文件锁内容为应用的进程号。

11.2 多用户单进程

多用户单进程情况下，应用宜使用文件锁的形式实现单进程，文件锁于`/run/应用名.pid`。

11.3 特殊访问单进程

特殊访问单进程情况下，应用宜使用文件锁的形式实现单进程，文件锁于`/run/user/用户 ID/应用名.pid`。

12 多进程要求

12.1 进程间通信要求

进程间可使用信号、管道、套接字、消息队列、信号量、共享内存等方式实现通信。

12.2 系统资源限制要求

系统在`<limits.h>`文件中设置默认的应用资源限制，每个进程都有一组资源限值。

每个资源都有软限制和硬限制 2 种关联限制，软限制规定了进程能够消耗的资源数量，进程可以将软限制设置为 0 到硬限制之间的值；硬限制为软限制设定上限，只有特权进程能够增大或者缩小硬限制。

进程可使用 `getrlimit()` 函数读取自身的资源限制，可使用 `setrlimit()` 函数修改自身的资源限制。

12.3 系统隔离要求

关键电网调度类应用宜使用 `cgroup` 对 CPU、内存、IO 等资源进行限制，防止资源竞争影响主控进程。

13 兼容性要求

除基础 C 库以外，应用不应依赖操作系统提供的任何动态链接库。

应用不应直接修改操作系统提供的配置文件，宜通过操作系统提供的接口来修改，可以通过新增高优先级配置文件来进行配置覆盖，新增的配置文件以“两位数字-应用名.conf”的形式命名。

应用软件应充分考虑国产 CPU、操作系统、数据库、中间件、浏览器等信创环境的兼容性要求，避免对单一技术路线、特定操作系统发行版、专有接口或特定动态库形成强依赖。

附录 A
(资料性)
变更日志文件格式

1、deb 包变更日志 changelog 文件的格式定义如下：

package (version) distribution (s) ; urgency=<选项值>

* change details more change details

* even more change details

-- maintainer name <email address>[two spaces] date

其中，package (version) 是安装包的名字和版本号；

distribution (s) 代表要发行的软件版本在上传时需要安装在哪里；

urgency 是为了表示这个版本比上一个版本的重要程度，它的选项有 low, medium, high, emergency, critical;

change details，至少要有 2 个空格开始，而且要以*号和分隔符[一般是空格]开始，下一行要和上一行对齐，建议这里不要写太多；

maintainer name，是安装包的维护人员的名字；

Email，邮箱地址；

Date (day-of-week,dd month yyyy hh:mm:ss+zzzz)，时间。

2. 变更细节 change details 可记录的选项包括：

1) 新增 (Features)：新增功能；

2) 修复 (Fixed)：修复 bug；

3) 变更 (Changed)：对某些已存在功能所发生的逻辑变化；

4) 优化 (Refactored)：性能或结构上的优化，并带来功能的逻辑变化；

5) 即将删除 (Deprecated)：不建议试用或在后续版本中即将删除的功能；

6) 删除 (Removed)：已删除的功能。

3、RPM 包变更日志文件的格式定义如下：

第一行：* 星期 月 日 年 修改人电子信箱 - 修改后的版本号

第二行：- 对哪些内容进行了修改，可写多行，一般以减号开始，便于后续的查阅。

附录 B
(资料性)
desktop 文件格式

desktop 文件的参考格式如下:

[Desktop Entry]	
Version=1.31.38.0	#版本
Name=AnXinTong	#英文名称
Name[zh_CN]=即时通讯	#中文名称
Comment=AnXinTong	#软件英文注解
Comment[zh_CN]=即时通讯	#软件中文注解
Icon=cpk.akey.chat	#icon 名，不带路径需要将 icon 安装至系统默认图
标路径下，否则需绝对路径	
Exec=/opt/AnXinTong.sh	#软件运行文件绝对路径
Terminal=false	#是否使用终端
Type=Application	#启动器类型
Categories=Application	#应用类型

附录 C
(资料性)
符合性调研表

检查项类别	检查内容	要求要点	是否符合
编码语言	开发语言使用情况	使用主流语言（如 C++17、Java11、Python3.8 及以上）	☐ 是 ☐ 否
命名规范	是否统一命名方式	使用驼峰/匈牙利/帕斯卡/下划线之一，并制定命名约定文档	☐ 是 ☐ 否
变量命名	是否规范命名变量	避免缩写、保持清晰，临时变量仅限循环中使用	☐ 是 ☐ 否
国际化支持	是否支持 UTF-8/BIG5 编码	不得硬编码中文字符，应使用 .ts/.qm 等格式管理多语言	☐ 是 ☐ 否
变更日志	是否有版本日志文件	每版本对应一条变更日志，符合标准附录 A 格式	☐ 是 ☐ 否
存储位置	可执行文件位置	安装路径位于 /bin、/usr/bin、/opt/应用名/bin 等标准位置	☐ 是 ☐ 否
存储位置	配置文件存储位置	全局/etc/应用名/，用户/home/.config/应用名/	☐ 是 ☐ 否
存储位置	数据与缓存目录	数据放置于 /var/lib 或 /opt/应用名/data；缓存置于 /var/cache 等	☐ 是 ☐ 否
日志输出	日志规范性	INFO 不在循环体中输出，结构化（JSON）格式，统一日志级别	☐ 是 ☐ 否
日志存储	日志保存位置与时间	默认目录为 /var/log/应用名，至少保存 15 天	☐ 是 ☐ 否
安全运行	运行权限控制	不允许 sudo/pkexec 提权，前后端分离，使用 dbus 通信	☐ 是 ☐ 否
凭据管理	密码存储方式	不允许明文存储，应加密或使用密钥管理系统	☐ 是 ☐ 否
网络安全	通信加密机制	前后端通信应启用 TLS 加密	☐ 是 ☐ 否
安全审计	是否记录敏感行为	敏感操作应记录审计日志	☐ 是 ☐ 否