GLMS

国联民生证券股份有限公司企业标准

Q/GLMS J002-2026

应用系统日志管理标准

Application system log management Standards

2026-06-11发布　　2026-06-11实施

国联民生证券股份有限公司 发布

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。

本文件参考《证券期货业经营机构内部应用系统日志规范》，并结合公司实际投产情况编写。

本文件起草单位：国联民生证券股份有限公司。

本文件主要起草人：丁旦、陈松夏、丁安安、杨怀宇。

目　　次

国联民生证券应用系统日志管理标准

1.引言

为规范国联民生证券股份有限公司（以下简称“公司”）应用系统日志的全流程管理，确保日志数据在采集、存储、分析及归档环节的完整性、一致性与安全性，优化存储架构并提升日志分析效能，以达成系统稳定性提升、故障快速定位与数据安全合规的目标，根据国家相关法律法规、证券期货业标准规范及公司信息技术管理制度要求，特制定本文件。

本文件适用于公司所有信息系统的日志管理活动，涵盖自研系统、外购系统及第三方提供的云服务或托管系统，包括但不限于交易、风控、账户、清算等各类业务系统及相关技术支撑平台。公司各部门、分支机构及全体员工在从事与信息系统日志相关的设计、开发、测试、运维、使用及管理活动时，均应遵守本文件。

通过本文件的实施，我们期望能够为应用系统的日志管理带来秩序，为系统的稳定运行提供坚实的基础，为业务决策提供有力的数据支持，最终推动我司信息系统的整体进步和发展。

2.术语和定义

下列术语和定义适用于本文件。

2.1 应用系统 (Application System)

应用系统是由应用软件、系统软件、网络设备、安全设备以及应用软件所依赖的软件包构成的软、硬件资源的统称。它包括但不限于数据库、中间件、操作系统和硬件设施。

2.2 日志 (Log)

日志是计算机系统、设备、软件等在运行过程中按时间顺序记录的有序数据集合。这些数据集合用于监控、调试、安全审核和问题解决。

2.3 日志采集 (Log Collection)

日志采集是从信息使用者的需要出发，通过各种渠道和形式获取日志相关信息的过程。这个过程涉及日志的识别、提取和传输至集中处理系统。

2.4 日志转储 (Log Dump)

日志转储是将日志内容转换为另外一种更具可读性或更易于分析的方式的过程。这通常包括格式化、过滤和转换日志数据，以便于用户理解和使用。

2.5 敏感信息 (Sensitive Information)

敏感信息是指在日志中存在用户的个人信息以及交易相关的信息等。这类数据包括但不限于用户的隐私数据，如手机号、身份证号、银行卡号、密码、交易标的代码、名称、价格、数量等。

2.6 日志代理 (Log Agent)

日志代理是完成日志采集并向日志管理中心发送采集到的日志数据的功能组件。

2.7 日志管理中心 (Log Administration Center)

日志管理中心是由软件和硬件构成的对日志数据进行采集、集中存储、分析、查询、监控等处理的统一服务平台的统称。

2.8 日志分析 (Log Analysis)

日志分析是对收集到的日志数据进行处理和解释的过程，目的是发现潜在的问题、趋势或模式。

2.9 日志保留策略 (Log Retention Policy)

日志保留策略定义了日志数据应该保存多长时间，以及在什么条件下可以删除或归档。

2.10 日志审计 (Log Auditing)

日志审计是定期检查日志数据以确保符合安全政策和法规要求的过程。

3.日志设计

3.1 日志内容组成

日志内容组成需包含时间戳、日志级别、组件名（产生日志的类名、模块名或服务名）。其他日志元素，如进程/线程 ID、请求 ID/TraceID、业务关键参数、执行结果、错误详情等需根据实际需要进行添加。

3.2 日志输出结构

日志输出既可以用非结构化文本（传统格式）方式，也可以用结构化（如 json）方式。但是每条日志的输出格式需保持统一。

3.3 日志等级分级

日志需进行分级，分级类别依次为：

DEBUG：调试信息，用于在开发或测试环境中诊断复杂问题。生产环境必须关闭。

INFO：记录程序正常的运行状态和关键业务节点，用于跟踪系统主要流程。

WARN：警告信息，表示发生了意外但不影响核心功能的潜在问题。

ERROR：错误信息，表示某个操作失败，影响了当前请求，但应用整体仍可运行。

FATAL：致命错误，导致应用程序或系统完全无法继续运行，需要立即人工干预。

3.4 日志轮转逻辑

为了防止单个日志文件无限增大，占满磁盘空间，日志需具备轮转逻辑。一般为基于大小或基于时间进行轮转。

3.5 容器日志

容器云架构应用日志输出应尽可能采用标准输出，日志将进行平台层面的采统一采集，确因为应用需求需要输出到应用内部的，由平台与架构团队评估后专项进行采集和管理。

4.日志记录

4.1 基本要求

（1）运行记录：日志记录机制应设计为对应用系统和程序的常规运行不产生负面影响，确保业务流程的顺畅和稳定性。

（2）日志级别：应根据系统组件和业务需求，预设合理的日志记录级别。这包括为不同模块和功能配置适当的错误、警告、信息和调试级别。一般为 *ERROR*、*WARNING*、*INFO*、*DEBUG*。生产环境禁止开启 *DEBUG*。

（3）日志内容：日志应精确记录关键信息，避免冗余。

（4）记录频次：对于频繁发生的相同日志事件，应实施频率控制策略。若循环内记录日志，应避免异常时无限记录日志的可能，造成磁盘可用空间的耗尽。

（5）异步记录：考虑实施异步日志记录，将日志写入操作从主业务流程中分离，减少对主线程性能的影响。

（6）日志轮转：日志应实施智能日志轮转，确保单一日志文件不会无限增长。

4.2 内容要求

（1）日志易读性：日志记录应设计为易于人类阅读和机器解析，采用清晰的结构和直观的格式。

（2）日志格式：各个应用系统、系统下的子服务，格式应统一，对于结构化的日志，建议使用'日期 时间 日志等级 其他内容'的模板。对于非结构化日志，应使用明确的、唯一的分隔符。

（3）日志编码：所有日志记录应采用 UTF-8 编码，以支持国际化字符。

（4）日志分类：应用系统运行日志、业务日志、审计日志等应分别独立记录。

（5）日志脱敏：对于包含敏感信息的交易日志，如客户交易流水，必须进行脱敏处理，若原始日志脱敏存在后期调查取证问题，可借由日志平台对转储后的日志进行脱敏。此外，应将其与其他业务日志分开存储，并实施严格的访问权限控制，以确保数据安全。

4.3 完整性要求

（1）日志行独立：每条日志应独立成行，避免使用换行符进行跨行记录。若无法避免跨行，应保证首行标识的一致性，例如'2025-01-01'，日志代理需据此标识合并日志。

（2）日志长度：单行日志的长度应不超过 4KB。若单行日志内容超出此长度，应精简记录，仅捕捉和记录日志的核心要素。

（3）日志完整性：在多线程环境下，确保日志系统的设计能够防止日志被截断、覆盖或丢失，保障每条日志的完整性。

（4）日志追踪性：对于需记录访问链路的日志，应具备可追踪性，类似于日志中嵌入同一 traceid。

（5）日志原子性：确保日志记录操作的原子性，即每条日志记录要么完整写入，要么完全不写入。

（6）日志保存：日志应根据审计要求保留适当期限，日志应定期压缩归档，存放于归档存储上或收集存入日志中心，日志中心根据监管需求设置保留期限。

4.4 一致性要求

（1）同一应用系统日志文件命名格式和风格应保持一致。

（2）同一应用系统日志字段命名格式和风格应保持一致。

（3）同一应用系统同一业务类型日志级别应保持一致。

（4）在数据采集等操作时应验证数据与日志源数据一致。

（5）日志格式的改变应遵循统一，所有子系统与服务应保持一致。

4.5 记录要求

（1）宜记录场景：

①应用系统启动及退出时，宜在日志中记录启动和退出过程。

②应用系统重要的启动配置。

③应用系统所有的错误信息。

④应用系统所有的警告信息。

⑤接口调用较长时间等待。

⑥重要的业务状态变化。

⑦定时任务启动和执行情况。

⑧客户端访问日志。

⑨耗时程序的执行进度。

⑩其他监管需要的记录场景。

（2）不宜记录场景：

①避免在循环体多次迭代中记录日志。若需要记录日志，则应设置达到合理的迭代次数之后再输出一次日志。

②避免在日志中记录过大的消息或内容。

4.6 日志分类

应用日志应至少包含以下四大类:

（1）操作系统日志

操作系统日志除应记录基本内容外，还宜记录以下信息：操作系统的启动、关闭信息；用户登录、退出信息；权限变更信息；系统运行状态信息，包括内存使用率、CPU 占用率、磁盘使用情况等；主机系统服务或配置变更信息。

（2）运行日志

运行日志记录了系统运行时的各类信息，包括但不限于应用的启动和关闭、错误和异常、服务的运行状态、服务的配置加载信息、资源的调用信息等。

（3）审计日志

审计日志基于安全和合规目的，记录了需要跟踪的审计活动，包括但不限于用户登录验证信息、访问控制、操作审计、安全事件、配置策略变更等。

（4）业务日志

应用系统业务相关的日志，主要包括用户行为日志、业务行为日志、交易日志（如涉及）和诊断日志）。

用户行为日志：记录用户操作行为的日志。用户行为日志信息包括但不限于用户标识、登录时间、登录 IP 以及系统操作日志等。

业务行为日志：记录应用系统内部业务处理相关的日志，通常为业务处理的逻辑描述、状态和结果的留痕信息。此类日志包括业务标识、业务处理耗时、业务请求参数、处理状态及处理结果等信

息；

交易日志：交易日志应记录业务流水、交易报文等信息，可用于分析系统的业务特征，如交易量、活跃客户数、交易流动性等，以及用户异常交易行为监控及交易异常问题排查等。此类日志包括用户标识、交易业务参数、交易状态及结果等信息，涉及敏感信息应进行脱敏处理；

诊断日志：记录应用系统的异常信息，用于问题排查和监控。此类日志应记录异常发生的出处信息、详细异常信息以及异常堆栈信息等。

4.7 日志命名

（1）日志文件名称中应包含应用标识、日志级别，日志归档文件应加上时间标识。

（2）同一应用系统内日志命名方式和风格保持统一，同一应用系统内日志文件名应具有唯一性。

（3）如果日志文件以时间分，建议每个日志文件以该日志文件的起始时间作为编号，时间采用 24 小时制。

（4）同一应用系统日志字段含义应统一和唯一。

（5）同一应用系统同类日志，字段种类与顺序保持一致。

5. 日志容灾归档

（1）日志的多副本策略统一由日志平台提供，在常规运行情况下，日志平台提供 1 副本的容灾能力。

（2）在日志保留期内，任何人不应对所有的原始日志和记录进

行更改、删除等操作。

（3）日志数据应建立归档机制。对于具备一般保存时效要求的应用系统，日志默认保存周期为 6 个月；网络监控日志、重要信息系统监控日志保存周期 1 年，以满足等级保护相关要求，期间日志平台提供在线存储与查询服务。针对有长期保存需求的系统（如重要信息系统业务日志保存 5 年及以上等），日志平台将统一提供转储至 NAS 磁盘并执行压缩存储的能力。

6. 日志采集

6.1 采集方式

日志采集统一由公司日志平台负责。传统日志、paas 服务日志默认通过日志平台 agent 采集器进行采集。亦可选择 syslog 推送方式。容器日志一般由容器平台统一 syslog 转发至日志平台。

6.2 采集频率

日志采集一般情况下为实时采集，部分特殊敏感系统可选择采集时间段采集，但需评估日志延迟和集中采集压力问题。

6.3 采集标准

（1）日志采集不应影响应用系统的正常运行。

（2）日志采集应对用户网络和业务的影响最小化。

（3）日志在不可信任的环境中进行传输时应加密处理，避免中间过程截取日志。

（4）日志应经过授权才可采集和使用。

（5）未经允许，证券期货机构外的第三方应用系统和机构不得

以任何形式采集、访问和下载业务系统日志。

（6）日志提供方应明确日志的安全保护要求，并明确告知数据接受方。

（7）日志接收方对于获取的不同应用系统的日志进行汇聚时，应满足安全保护要求。

7. 日志运营

（1）接入日志平台的应用，日志查询能力将统一由日志平台提供。

（2）对有统计需求的日志，需解析并提取关键字段。应用方提出需求后，由日志平台管理员统一创建解析规则。

（3）日志统计需求由应用方提出，由日志平台管理员实施后交付。对于交付后的仪表盘，应用方具备编辑权限。

（4）日志告警需求由应用方提出，由日志平台管理员实施后交付。对于交付后的告警规则，应用方具备编辑权限。

（5）日志告警方式原则上统一发送至告警中心，由告警中心再下发至各告警通道。

（5）针对现有日志的数据挖掘，须以业务价值与风险洞察为目标。应用协同日志平台从合规存储的日志中提取模式、趋势与异常进行统计分析，并持续评估有效性。

8.日志权限管理

8.1 本地日志文件权限管理

（1）最小权限原则

身份隔离：应用进程与服务应运行于专属的低权限账号之下，仅具备其所需日志目录的写入权限，杜绝权限滥用。

用户权限分离：系统用户应依据职责被授予差异化权限，绝大多数用户账户不应具备直接读取系统日志的能力。

（2）访问控制

通过文件系统的自主访问控制列表实施管控。建议将日志目录权限设置为 755，日志文件权限设置为 640（所有者读写，所属组只读，其他用户无权限）。对于包含敏感信息的日志，权限应更为严格，设置为 600（仅所有者读写）。

8.2 集中式日志平台权限管理

（1）统一身份登录控制

日志平台已完成与公司统一认证基础设施——飞连系统的全面对接。根据安全管理原则，所有用户必须使用飞连账号体系进行身份验证登录日志平台，原则上禁止使用自建账号登录。

（2）基于角色的访问控制

日志平台采用“用户-分组-角色-资源”的多层级权限管理体系。为确保日志数据的安全性与隔离性，所有接入系统的日志资源分配与角色权限配置，均由平台管理员统一实施强制访问控制。核心策略是：严格限定各角色仅可检索其自身授权系统的日志数据，

从机制上杜绝跨应用日志的越权访问。

9.日志资源管理

9.1 应用运维方

采集维护：负责日志输出端的全生命周期管理，包括确保应用正常产生日志，以及日志采集器的安装、部署。

9.2 日志平台管理方

平台维护：负责日志平台集群自身服务的部署、监控、高可用、升级与故障处理。

配置管理：负责日志采集任务、日志统计分析、告警规则等核心功能的配置与管理。

9.3 基础设施团队

资源保障：负责保障日志平台所依赖的主机、网络、存储等底层基础设施的稳定运行。

资源扩容：根据平台管理方的需求，执行主机、网络等资源的扩容操作。

9.4 DBA 管理员

数据库管理：负责日志平台后端 OceanBase 数据库的安装、配置、监控、性能优化、备份与容灾管理。

10.日志准入

（1）日志记录机制应设计为对应用系统和程序的常规运行不产生负面影响，确保业务流程的顺畅和稳定性。

（2）日志中需包含日志级别，级别一般为 ERROR、WARNING、INFO、DEBUG。仅允许在查找生产问题、灰度发布和试运行等特殊场景下可开启 DEBUG,并及时关闭。

（3）日志文件需实施日志轮转，确保单一日志文件不会无限增长。

（4）日志首行以日期（“YYYY-MM-DD”）、时间开头。

（5）日志记录须保证原子性。

（6）日志文件名称中应包含应用标识、日志级别，日志归档文件应加上时间标识。

（7）同一应用系统同类日志，字段种类与顺序保持一致。

（8）敏感信息需脱敏，理论上交易类、个人信息类（身份证、密码等）日志必须进行脱敏。

（9）同一应用系统日志应设置为统一的存储路径，可根据服务类型不同设置不同子目录进行存储。

（10）在循环语句中应避免嵌入日志打印逻辑，以防止因循环结构异常而引起日志输出失控或系统性能下降。

11. 日志审计

为保障应用系统日志管理的规范性与有效性，可结合实际审计需求，参照以下要求开展相关工作：

（1）审计过程中可关注应用系统日志记录的准确性、完整性、安全性及合规性，对日志的记录、转储、存储、备份、销毁等全流程，以及重要用户行为、异常操作、重要系统命令使用等重点内容进行核查，必要时可形成相关报告。

（2）可根据审计工作需要，提供必要的支持保障，包括但不限于所需的系统访问权限、日志 数据、文档资料，以及适宜的工作场所、网络资源等；对审计相关的问询与反馈及时回应，按需补充相关信息或作出解释。

（3）可结合审计结果与建议，适时优化日志管理及内部控制流程；审计全过程中关注相关法 律法规与行业标准的遵循，审计结束后妥善留存审计文档及沟通记录，为后续工作提供参考。

12. 附则

（1）本细则由信息技术总部负责解释。

（2）本细则自发布之日起生效。

（3）信息技术总部将至少每年一次对本办法的执行情况进行回顾和评估，结合技术发展、监管要求变化、业务需求及内部审计发现，对其进行必要的修订和完善。

（4）信息技术总部平台团队负责组织对本办法的培训和宣贯，

确保所有相关部门和人员理解并遵守其规定。新员工入职培训应包含本办法相关内容。

附录-日志实例

基于公司内常见的日志使用场景，例如java、nginx、docker等，附录将给出指导性的配置标准。

JAVA 样例

```
2024-08-15 15:08:17.696 ERROR [app-counter-all-web] [http-nio-18843-exec-10] [com.mszq.cfh.appcounterall.web.config.mvc.TokenInterceptor] [traceId=T1cbc7d173f4944fe9c235ae5dfe86c28] [line:60] handleToken | token不存在 |
4;192.168.178.121;OPPO;;;13163247039;b6f6f54b34fc17d0;14;218.17.161.51,0;;;b6f6f54b34fc17d0;TDX_GPHONE;4.11;2024-05-29 16:33
```

字段解释：

【时间戳】完整的时间戳，包括日期和时间，精确到毫秒。

【日志等级】日志的严重程度，常见的有…… 。

【应用名称】应用的名称，用于区分不同的应用。

【线程名】记录日志的线程名称。

【类名或模块名】记录日志的类名或模块名。

【trace ID】用于追踪请求的唯一标识。

【代码行号】记录日志的代码行号。

【方法名】记录日志的方法名。

【日志消息】具体的日志消息。

【附加信息】其他自定义的附加信息，可以根据实际需求进行调整。

logback 配置

```
<configuration>
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %level [%app] [%thread] [%logger] [traceId=%X{traceId}] [line:%L] %method | %msg | %ex%n</pattern>
        </encoder>
    </appender>

    <root level="info">
        <appender-ref ref="STDOUT" />
    </root>
</configuration>
```

SLF4J 和 Log4j2 配置

```
<Configuration status="WARN">
    <Appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSS} %level [%app] [%thread] [%logger] [traceId=%X{traceId}] [line:%L] %method | %msg | %ex%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Root level="info">
            <AppenderRef ref="Console"/>
        </Root>
    </Loggers>
</Configuration>
```

NGINX 样例

```
192.168.1.1 - - [15/Aug/2024:15:08:17 +0800] "GET /index.html HTTP/1.1" 200
```

1234 "http://referrer.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36" "192.168.1.2" 0.003 0.002

nginx 配置

```
$remote_addr - $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http_user_agent" "$http_x_forwarded_for" $request_time $upstream_response_time
```