Android 15 日志
使用指南

版本号: 2.0
发布日期: 2025.03.21

版本历史

版本号	日期	制/修订人	内容描述
1.0	2023.03.02	AWA1830	初始版本文档
2.0	2025.03.21	AWA2290	AwLog 2.0 版本文档

目　录

插　图

1 AwLog 2.0 使用指南

1.1 简介

AwLog 2.0 是全志研发 AwLog 1.0 的升级版本，不但保留了原有功能，而且支持收集 uboot 阶段的日志和提供更多便捷的亮点功能。

1.1.1 适用平台

Android V 及以上 Aw 的 Android 平台。

1.2 日志类型和参数

日志类型主要有 Android Logcat、Uboot、Kernel、System Info 等，对应可配置日志开关、单文件最大大小、文件最大数量等参数，如下图，其中加粗为默认配置。

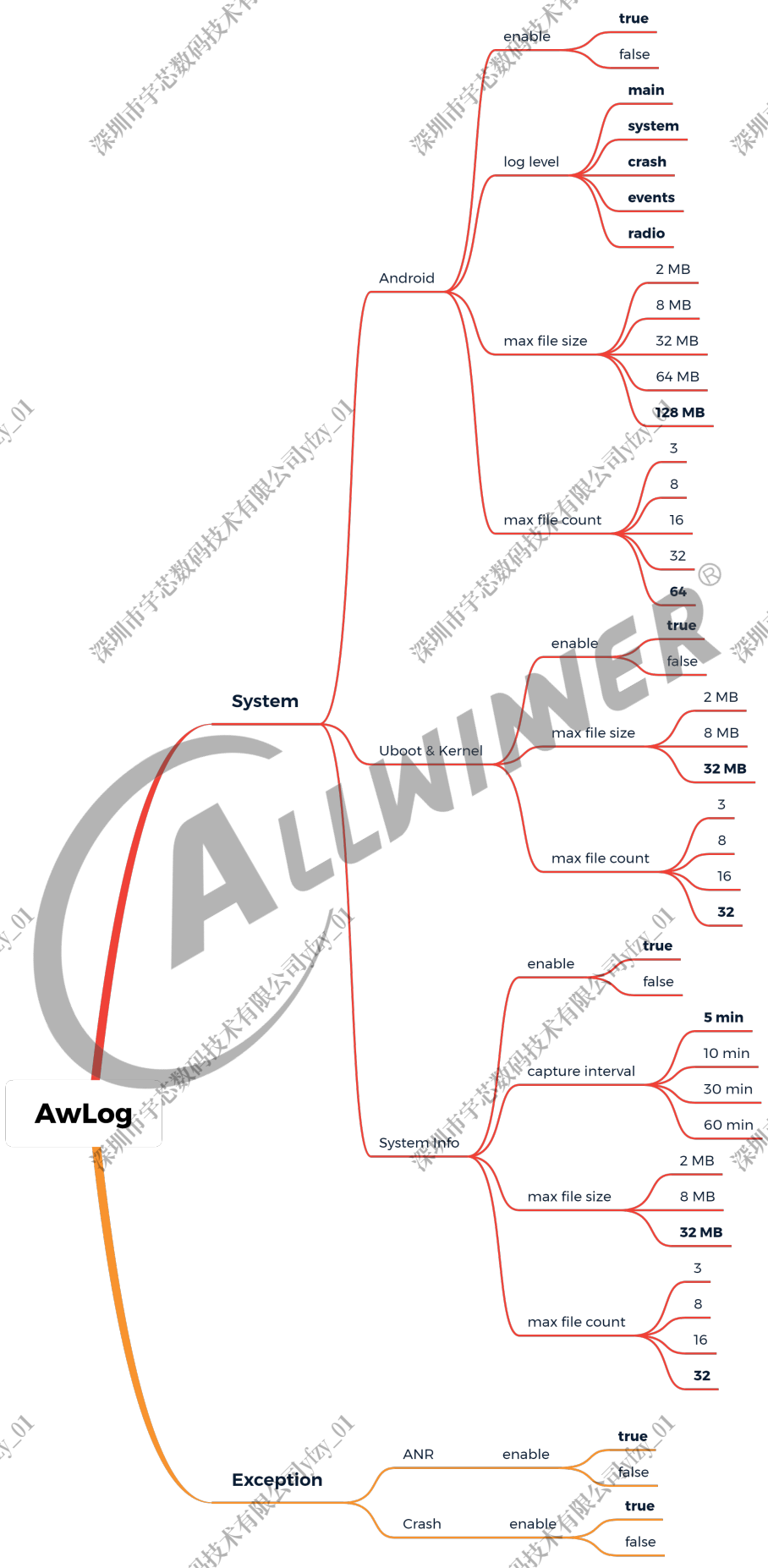

图 1-1. 日志类型和参数

1.3 日志场景

日志场景是日志类型和参数设置的组合，除默认场景外用户可以在 主页 -> 场景 -> 右上角 “+” 点击进入新增场景。例如，只需要抓 Uboot 和内核日志，单文件最大 8 MB，最多 16 个文件，可以按如下配置保存。

图 1-2: 日志场景

默认场景不可编辑，自定义场景提供编辑、删除等功能，点击场景可切换当前场景。

图 1-3: 场景列表

1.4 日志开关

在主页有展示当前场景抓取的日志类型，打开左上日志开关就会按照当前场景配置的参数抓取日志，并显示已抓取时间。

图 1-4: 日志开关

总开关的设置和获取也可以通过如下属性：

Property	Type	Default	Description
persist.debug.logpersistd	Boolean	true	总开关

1.5 日志记录

点击 主页 -> 历史 进入到日志记录，这里对每次抓取的日志按时间段保存，点击记录可展开详情。点击记录下文件名可打开浏览，右边从左至右分别是上传、导出、删除功能，具体将在后面介绍。

图 1-5: 日志记录

文件存储在 /data/media/awlog，由于用户无 data 权限，因此通过 sdcardfs 映射，可以通过 /storage/emulated/awlog 目录获取相应的 log。其中目录结构和日志记录一一对应，当日志文件大小超出限制会重新输出到新的文件，当文件数量超出限制会进行循环覆盖，新文件覆盖旧文件。如 logcat 单个文件过大，则分为 logcat、logcat.1、logcat.2 …

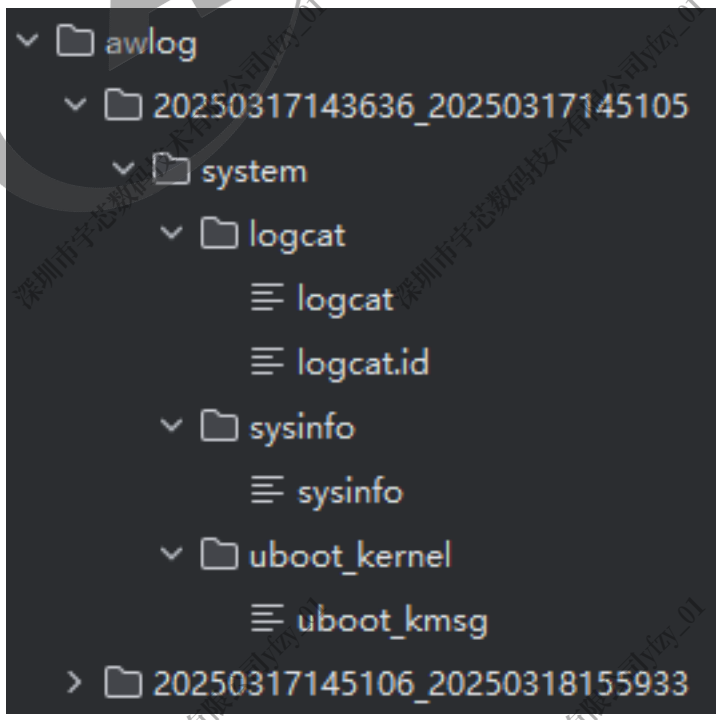

图 1-6: 文件存储

1.6 导出日志

在日志记录里面点击导出按钮，弹出选择导出路径，可选择内部存储或连接的可移动存储，确定后将该记录的文件打包成压缩包保存至指定路径。

图 1-7: 导出日志

也可以点击 主页 -> 导出，这里的区别在于会导出最新的日志，如目前未在抓取则导出已有的最新日志。如目前正在抓取，会先停止抓取，导出这份日志，再开启一次新的抓取。

图 1-8: 导出最新日志

1.6.1 获取全部日志

开发人员可以通过 adb pull /data/media/awlog 命令直接将全部日志提取到电脑。

1.6.2 拼接日志

为了方便管理日志大小，日志通常由多个文件组成，可以通过以下命令将多个日志组合成单独的日志文件。

```
ls -r logcat* | xargs cat > log.txt
```

1.7 上传日志

在日志记录里面点击上传按钮，会弹出输入框，要输入服务器地址和文件字段名，其默认值读取如下属性。

Property	Type	Default	Description
persist.debug.upload.url	String	-	服务器地址
persist.debug.upload.key	String	-	文件字段名

上传完成服务器会收到日志压缩包，此外，可在 Header 的 Device 字段中获取 SDKVersion-Name、SDKVersionCode、RomInfo 等信息。

图 1-9: 上传日志

1.8 预览日志

在日志记录里面点击文件名可打开文件进行预览，提供搜索关键字功能。点击放大镜按钮打开搜索栏，再次点击触发搜索，长按关闭搜索栏。

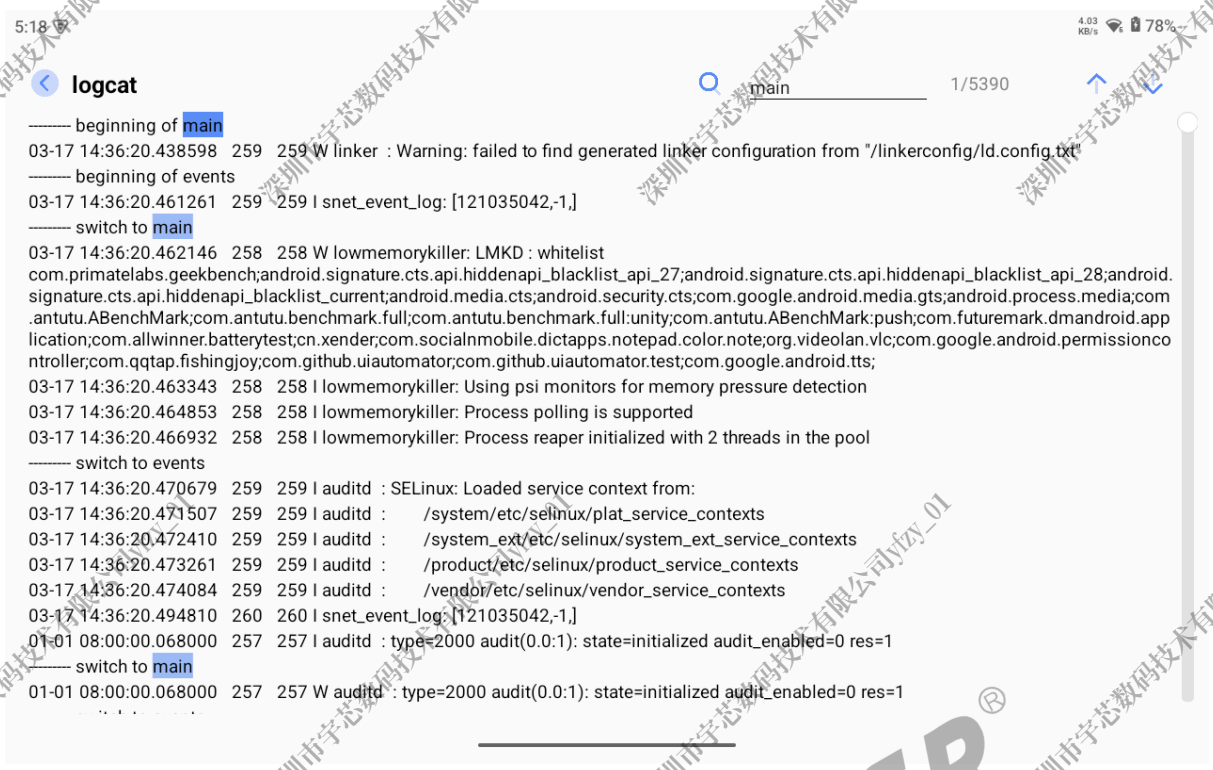

图 1-10: 预览日志

1.9 开机自启动

开机自启动首次启动默认打开，后续同步日志总开关状态，如关机时总开关打开，则开机自动开始抓取，反之则不会。开机自启动会继续写入上次开机未结束的记录中，或者创建一次新的记录开始写入。

1.10 重置

点击 主页 -> 重置，可将场景、设置初始化为默认值，并将停止抓取日志。

1.11 清除

点击 主页 -> 清除，可将所有日志记录及文件删除，并将停止抓取日志。

2 AwLog 1.0 使用指南

2.1 简介

此功能通过开机启动后台日志抓取进程，实时保存内核及 Android logcat 到存储中。当机器死机、掉电重启等现象后仍能获取之前保存的日志进行分析；同时也方便测试人员较为简单的获取必要的日志信息。

2.1.1 适用平台

- Android Q 及以上 Aw 的 android 平台。

2.2 开关配置

2.2.1 静态配置

在方案目录下的配置文件中，添加以下配置打开日志保存功能。

```
PRODUCT_DEBUG := true
```

2.2.2 动态开关

非 user 固件，可以通过 adb shell 进行设置属性，打开/关闭日志保存。

property	value	default	description
persist.debug.logpersistd	bool	-	总开关
debug.logcat.enable	bool	-	自动保存 logcat 开关，当次启动有效
persist.debug.logcat.enable	bool	-	每次开机后设置 debug.logcat.enable
persist.debug.logcat.buffer	string	default	需要保存的 logcat buffer 类型
persist.debug.logcat.size	int	32768	单个 logcat 文件大小
persist.debug.logcat.limit	int	8	logcat 文件个数（自动删掉最旧的）
debug.kernel_log.enable	bool	-	自动保存内核 log 开关，当次启动有效
persist.debug.kernel_log.enable	bool	-	每次开机后设置 debug.kernel_log.enable

property	value	default	description
persist.debug.kernel_log.size	int	8192	单个内核 log 文件大小
persist.debug.kernel_log.limit	int	8	内核 log 文件个数（自动删掉最旧的）
debug.crashdump.enable	bool	-	使能 crashdump 功能开关，当次启动有效
persist.debug.crashdump.enable	bool	false	每次开机后设置 debug.crashdump.enable

注：

1. persist.debug.logpersistd，persist.debug.logcat.enable，persist.debug.kernel_log.enable 根据方案配置 PRODUCT_DEBUG := true 设置为 true。
2. persist.debug.logcat.buffer 取值为 logcat buffer 取值，包含 main,system,radio,events,crash 中的一项或多项，使用 “,” 进行分隔，默认为 main,system,crash。

2.2.3 计算器开关

在计算器集成了动态开关设置，计算器中输入 “log(666+!)++” 弹出开关设置界面，根据 GUI 进行配置 log 保存选项。

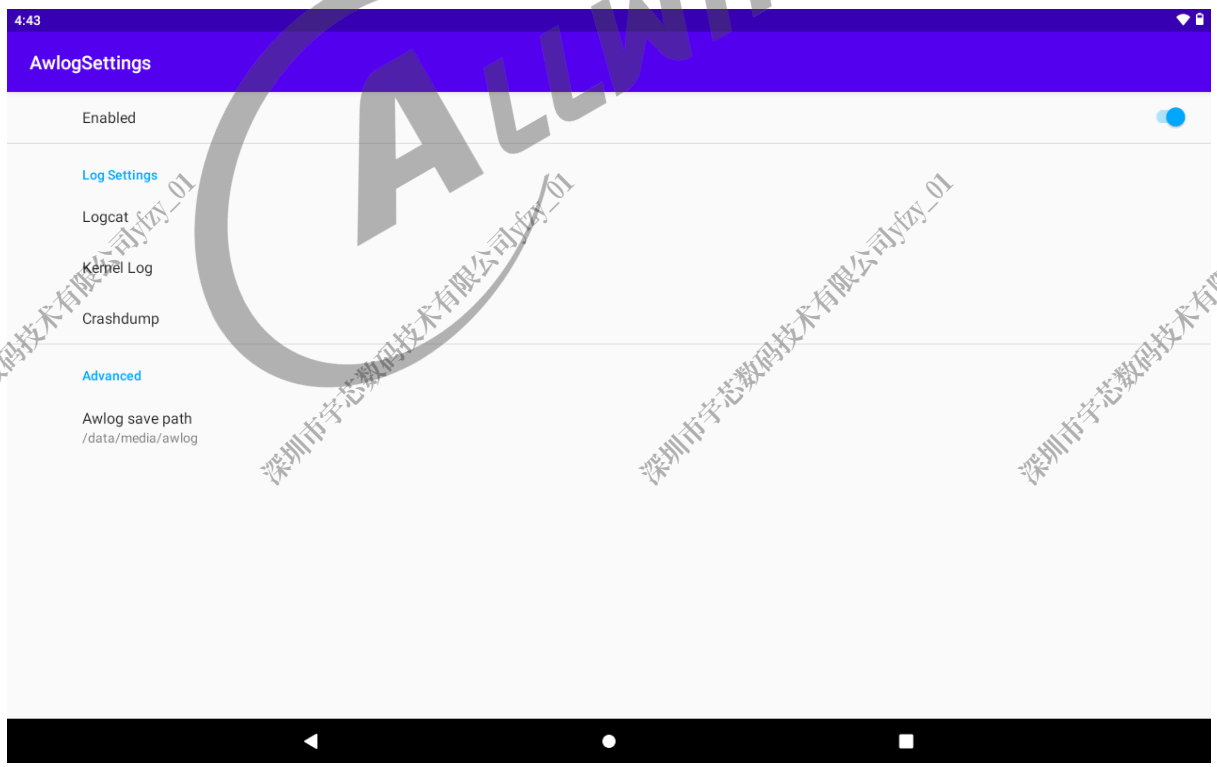

图 2-1: AwlogSettingsShot

注：

1. 其中 logcat 和 kernel log 的 Boot enabled 默认没有选择，配置时需要先选择好配置，最后再选择打开对应的 enbale 选项。

2. 其他配置选项，可根据下面思维导图，以及上一章节的属性介绍，自行选择是否配置。

思维导图如下：

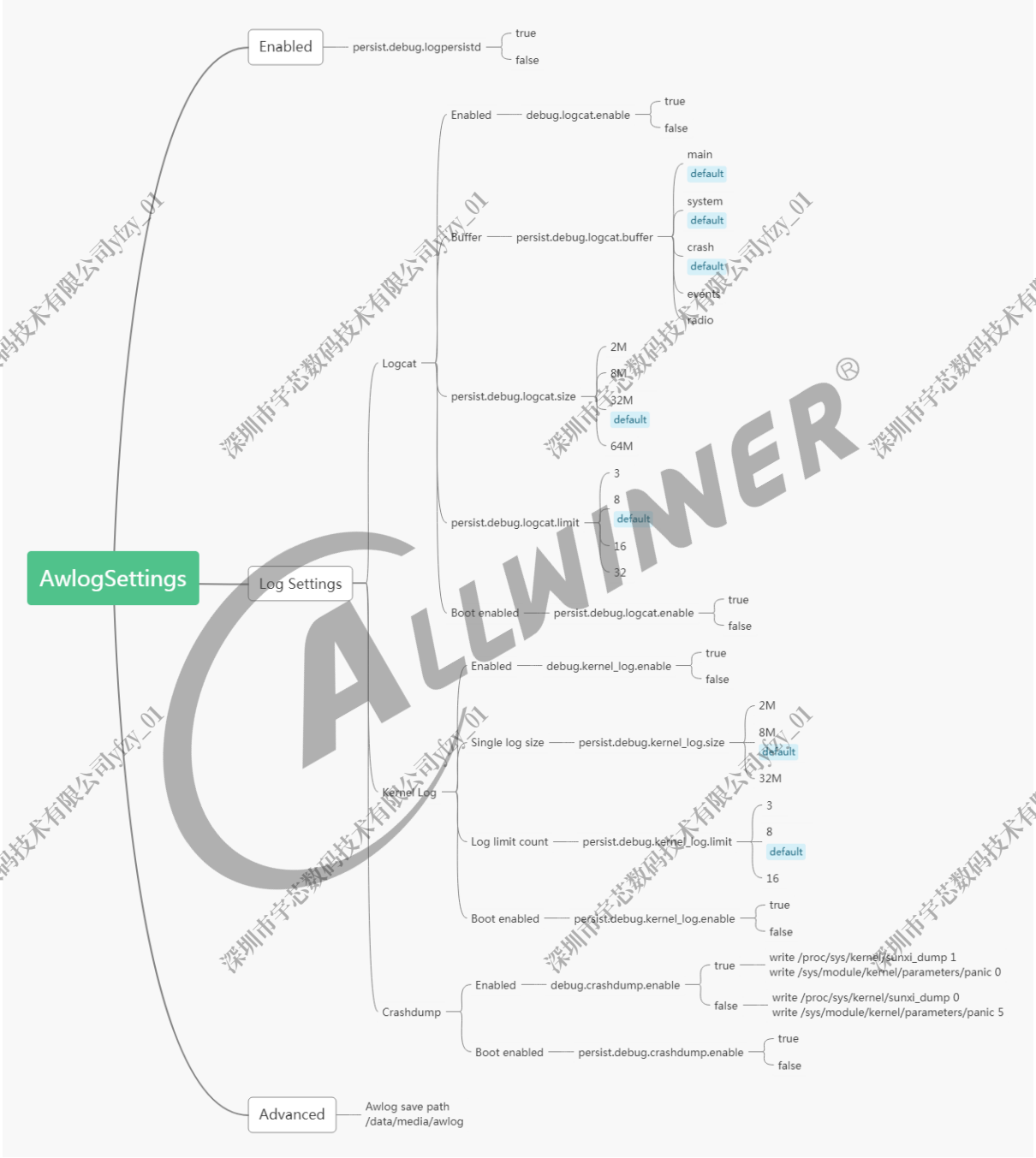

图 2-2: AwlogSettings

2.3 日志组成

2.3.1 目录结构说明

```
/data/media/awlog               # awlog目录
/data/media/awlog/logcat        # 存放logcat目录
/data/media/awlog/logcat/logcat     # logcat文件，如单个文件过大，则分为logcat、logcat.001、logcat.002、logcat.003等(序号越小越新)。
/data/media/awlog/kernel        # 存放内核log目录
/data/media/awlog/kernel/kmsg       # 内核log，如单个文件过大，则分为kmsg、kmsg.001、kmsg.002、kmsg.003等(序号越小越新)。
```

2.3.2 bugreport

日志会保存在 bugreport.zip 文件中，对应路径为 bugreport.zip/FS/data/media/awlog

2.3.3 用户目录映射

```
/storage/emulated/awlog -> /data/media/awlog
```

由于用户无 data 权限，因此通过 sdcardfs 映射，可以通过/storage/emulated/awlog 目录获取相应的 log。

2.4 如何获取日志

2.4.1 开发人员

开发人员可以通过

```
adb pull /storage/emulated/awlog
```

或

```
adb bugreport .
```

命令直接将日志提取到 PC 电脑。

注：旧版本 adb 可能不支持 bugreport，请更新最新版本的 adb 工具。

2.4.2 日志拼接

由于为了方便管理日志大小，日志通常由多个文件组成，可以通过以下命令将多个日志组合成单独的日志文件。

```
ls -r logcat* | xargs cat > log.txt
```

2.5 日志上传

图 2-3: UploadReportFragment

2.5.1 UploadReportFragment

app/src/main/java/com/softwinner/awlogsettings/fragment/UploadReportFragment.kt 调用上传接口，以及传入参数

NetRequest.buildJSON(str: String)

- post 请求，上传描述 bug 及其相关信息，构建初始表单

NetRequest.buildZIP(file: File, id: Int)

- form 表单，用于上传压缩包（收集的 crash 日志）,file 为 zip，id 为构建的表单唯一 ID。

2.5.2 NetRequest

在 app/src/main/java/com/softwinner/awlogsettings/net/NetRequest.kt 根据实际搭建的服务器接口进行修改相关参数如下。

```
private val KEY_ID = "" //项目id
private val ID = 1

private val KEY_SUBJECT = "" //项目名字
private val SUBJECT = ""

private val KEY_HEAD_TOKEN = "" //服务器权限凭据
private val HEAD_TOKEN = ""

private val CONTENT_TYPE = ""  //文件类型，json或zip
private val FORMAT_JSON = ""
private val FORMAT_ZIP = ""

private val REQUEST_URL = ""  //url

private val KEY_DESCRIPTION= "" //详细描述信息
private val KEY_FILE = ""   //文件名
private val KEY_ISSUE = "" //概要信息
```

如实际服务器中密钥 Id 为 key_id，则：

```
private val KEY_ID = "key_id"
```

著作权声明

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明


、 
、全志科技、  （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。