|  |  |
| --- | --- |
| 宝能汽车有限公司 | 编号： |
| 文档名称 | 版本：V1.0 |



**UA集成操作手册-mkotapackage**

|  |  |  |  |
| --- | --- | --- | --- |
| 信息分类 |  | 涉密等级 |  |
| 责任部门 | 智能网联研究院 | 责任人 |  |

**会签页**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文件编号：** | | **文件名称：** | | |
| **会 签** | | | | |
|  | **部门** | **姓名** | **签字** | **日期** |
| **编制：** |  |  |  |  |
| **校对：** |  |  |  |  |
| **审核：** |  |  |  |  |
| **批准：** |  |  |  |  |
| **各专业部门**  **会签：** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | **属性** | **属性负责人** | **签字** | **日期** |
| **发布日期：** |  | **版本** |  | |

**修订记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **作者** | **修订内容** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 概述 1](#_Toc49414709)

[1.1目标 1](#_Toc49414710)

[1.2 术语约定 1](#_Toc49414711)

[2 做包工具概述 1](#_Toc49414712)

[2.1 mkotapackage支持的分区类型 2](#_Toc49414713)

[2.2 mkotapackage的运行环境 2](#_Toc49414714)

[2.3 mkotapackage内置的算法协议版本号 2](#_Toc49414715)

[2.4 mkotapackage输出的升级包类型 3](#_Toc49414716)

[2.5 mkotapackage做包超时时间 3](#_Toc49414717)

[2.6 mkotapackage日志log 3](#_Toc49414718)

[3 mkotapackage做包工具详细说明 3](#_Toc49414719)

[3.1 mkotapackage支持的做包模式 3](#_Toc49414720)

[3.2 运行mkotapackage 3](#_Toc49414721)

[3.2.1运行准备 4](#_Toc49414722)

[3.2.2 创建源包和目标包 4](#_Toc49414723)

[3.2.3 创建配置文件 4](#_Toc49414724)

[3.2.4 生成升级包 5](#_Toc49414725)

[3.3 mkotapackage 配置文件详细说明 5](#_Toc49414726)

[3.3.1 文件规则路径 5](#_Toc49414727)

[3.3.2 mkotapackage命令行参数 5](#_Toc49414728)

[3.3.3 mkotapackage主配置文件 5](#_Toc49414729)

[3.3.4 指定升级包名称 6](#_Toc49414730)

[3.3.5 配置分区 6](#_Toc49414731)

[3.3.6 升级包类型 6](#_Toc49414732)

[3.3.7 版本信息 7](#_Toc49414733)

[3.3.8 升级包压缩类型 8](#_Toc49414734)

[3.3.9 配置分区信息 8](#_Toc49414735)

[3.3.10 分区名称 9](#_Toc49414736)

[3.3.11 分区类型 9](#_Toc49414737)

[3.3.12 分区文件系统类型 9](#_Toc49414738)

[3.3.13 分区设备节点号 10](#_Toc49414739)

[3.3.14 分区挂载点 10](#_Toc49414740)

[3.3.15 分区文件路径 10](#_Toc49414741)

[3.3.16 配置分区文件属性 10](#_Toc49414742)

[3.3.17 配置文件过滤器 12](#_Toc49414743)

[3.3.18 强制覆盖过滤器 13](#_Toc49414744)

[3.3.19 Not-In-Place方式升级 14](#_Toc49414745)

[3.3.20 升级只读文件系统 14](#_Toc49414746)

[3.3.21 配置分块升级 15](#_Toc49414747)

[3.3.22 SElinux支持 16](#_Toc49414748)

[3.3.23 内存优先支持 16](#_Toc49414749)

[3.3.24 优先调用本地升级程序 17](#_Toc49414750)

[3.3.25 设置可运行的最大并行线程数 17](#_Toc49414751)

[3.3.26 设置预升级功能 17](#_Toc49414752)

[3.3.27 定时器设置 17](#_Toc49414753)

[3.3.28 日志统计系统 17](#_Toc49414754)

[3.3.29 配置镜像文件自动挂载 18](#_Toc49414755)

[3.3.30 配置整版本校验功能 18](#_Toc49414756)

[3.4 主配置文件实例 19](#_Toc49414757)

[3.4.1 In-Place方式文件系统分区配置文件 19](#_Toc49414758)

[3.4.2 Not-In-Place方式文件系统分区配置文件 19](#_Toc49414759)

[3.4.3 非压缩类型升级包配置文件 20](#_Toc49414760)

[3.4.4 镜像分区升级配置文件 21](#_Toc49414761)

[3.4.5 镜像分区分块升级配置文件 21](#_Toc49414762)

[3.4.6 只读文件系统分区升级配置文件 22](#_Toc49414763)

[3.4.7 镜像自动挂载分区升级配置文件 23](#_Toc49414764)

[4 附录 23](#_Toc49414765)

[4.1 mkotapackage命令行参数 23](#_Toc49414766)

[4.2 mkotapackage配置文件参数说明 27](#_Toc49414767)

# **概述**

1.1目标

本文档介绍了mkotapackage做包工具，并详细介绍通过mkotapackage工具生成升级包的流程。

编写本操作手册的目的是为通过mkotapackage和Update Installer实现FOTA功能的项目开发人员，项目测试工程师和与项目相关的其他工作人员提供做包工具的使用指导。

1.2 术语约定

|  |  |
| --- | --- |
| **术语名词** | **说明** |
| In-Place方式升级 | 即有感升级，是指针对升级分区，分区的目标安装位置与源位置相同，可以理解为需要进入Recovery小系统对原分区进行升级。 |
| Not-In-Place方式升级 | 即无感升级，是指针对升级分区，分区的目标安装位置与源位置不同，可以理解为AB分区升级。 |
| 原始版本文件 | 零部件供应商根据源码编译产生的最原始版本文件 |
| 目标版本文件 | 零部件供应商根据源码编译产生的目标版本文件 |
| 整包 | 通过mkotapackage做包工具，将厂商发布的目标版本文件转化为可用于整包升级的特殊格式升级包 |
| 差分包 | 通过mkotapackage做包工具，将厂商发布的原始版本文件和目标版本文件通过算法差分运算，生成可用于增量升级的特殊格式升级包 |
| 升级包 | 用于描述用于升级的特殊格式的包，包括整包和差分包 |
| mkotapackage做包工具 | 用于制作标准格式升级包的工具 |

# 做包工具概述

mkotapackage做包工具（以下简称mkotapackage）是Linux系统下运行的命令行工具，用户通过命令行参数、主配置文件以及辅助配置文件生成标准格式的升级包。升级包的类型包含整包及差分包。整包内完整包含了用户发布的版本文件，差分包则包含了用户输入的源版本及目标版本的差分补丁文件，其大小取决于两个版本之间的差异与使用的mkotapackage内置算法选择。

2.1 mkotapackage支持的分区类型

mkotapackage支持的分区类型如下：

|  |  |
| --- | --- |
| **分区类型** | **详细类型** |
| 文件系统分区PT\_FS | EXT4/EXT2/UBIFS/DOS/QNX4/QNX6 |
| 非文件系统分区PT\_RAW | MTD/EMMC/NOR |

2.2 mkotapackage的运行环境

mkotapackage运行所需的运行环境如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **版本** | **描述** |
| 处理器 | 1.6GHz以上 | 建议值 |
| 操作系统 | Ubuntu 14.04以上 | 建议16.04 |
| RAM | 大于4GB | 建议值 |
| ROM | 大于源包和目标包的5倍 | 建议值 |
| 软件包 | Python3.5  libc++.so 1.0  mtd-utils | 3.5版本（或以上）  1.0版本  2.01版本（用于支持ubi文件系统做包） |

2.3 mkotapackage内置的算法协议版本号

在通过mkotapackage生成的升级包中，包含协议版本号（Protocol Version）信息。Protocol Version信息由算法产生，并通过mkotapackage将信息保存到升级包的head文件中。在通过Updater Installer安装升级包时，需先检测匹配Protocol Version后方可进行安装操作。

2.4 mkotapackage输出的升级包类型

mkotapackage支持生成zip压缩格式的升级包，同时也支持生成非压缩类型的升级包（bin形式），用户可通过mkotapackage配置文件进行配置。

## 2.5 mkotapackage做包超时时间

mkotapackage生成升级包时，可设置超时时间。当做包到达超时时间时，mkotapackage工具会退出做包过程。mkotapackage默认设置超时时间为5min，且用户可通过mkotapackage –T命令手动设置超时时间。

## 2.6 mkotapackage日志log

mkotapackage生成升级包过程会生成log文件，默认生成log文件路径为当前程序执行路径下log文件夹。

# mkotapackage做包工具详细说明

本章节详细介绍mkotapackage做包工具以及如何通过mkotapackage工具创建一个升级包。

## 3.1 mkotapackage支持的做包模式

mkotapackage工具支持以下几种做包模式：

1. 由目标版本生成整包：该模式需用户提供一个目标包，mkotapackage根据该目标包和配置文件生成一个整包，用于整包升级；

2. 由源版本和目标版本生成差分包：该模式需用户提供源版本包和目标版本包以及配置文件，mkotapackage根据源包和目标包，以及配置文件生成差分包用于差分升级；

3. 由两个整包（由mkotapackage通过方法①生成）生成差分包：该模式需用户提供两个通过mkotapackage生成的整包，mkotapackage根据两个整包生成差分包，用于差分升级。

|  |  |
| --- | --- |
| *NOTE* | 1. 可同时运行mkotapackage生成不同文件名的升级包，但不能同时运行mkotapackage生成同一个文件名的升级包，否则会产生未知的错误。  2. 影响mkotapackage做包时时间的因素：source与target的差异大小、运行mkotapackage的处理器性能、可用RAM、以及升级包的压缩类型。 |

## 3.2 运行mkotapackage

mkotapackage工具为linux系统下可执行二进制文件，可在linux shell下直接运行。mkotapackage通过源版本包和目标版本包生成一个升级包，包括整包或者差分包（整包不需要源版本），工具运行时需用户指定命令行选项、配置主配置文件以及辅助配置文件来完成指定功能：

* 命令行选项：mkotapackage工具命令行参数，请参考附录mkotapackage命令行选项。
* 主配置文件（XML文件）：做包前用户需配置主配置文件，用于配置生成升级包信息。
* 辅助配置文件：配合命令行、主配置文件完成mkotapackage指定功能，如过滤器配置文件（用于配置过滤功能，参考本文3.3.17章节）、属性文件（描述PT\_FS分区内文件属性，参考本文3.3.16章节）。

mkotapackage支持生成zip压缩格式的升级包，同时也支持生成非压缩类型的升级包（bin形式），用户可通过mkotapackage配置文件进行配置。

### 3.2.1运行准备

运行mkotapackage之前，需准备待升级的源包（整包升级不需要）以及目标包，同时满足以下要求：

1. 针对PT\_RAW类型分区，设置分块处理时，分块块数最大支持65535；

2. 针对Framsize （算法处理文件时逻辑扇区大小）可配置情况，Framsize在工具中默认最大值不超过10MB，最小值不小于4KB，推荐值为10MB以下；

3. 针对PT\_FS类型升级包，最大单个文件大小可超过4GB，但对内存、flash空间等做包环境要求较高；

4. 请控制一次做包生成的升级包大小，使其小于等于4GB（否则工具会报异常错误）。

### 3.2.2 创建源包和目标包

mkotapackage生成升级包以分区为单位，mkotapackage支持分区类型如下：

* PT\_FS：在配置文件中配置分区类型为PT\_FS，文件系统分区；
* PT\_RAW：在配置文件中配置分区类型为PT\_RAW，无文件系统分区；

一个升级包可以包含一个或者多个分区内容， 运行mkotapackage之前，需用户根据实际升级需求准备源版本和目标版本。版本文件以分区为单位提供，其中针对PT\_RAW类型分区可直接提供分区文件，针对PT\_FS类型分区，可对其分区内文件进行压缩（支持压缩类型：zip、tar.gz、tgz，且压缩包解压后为根路径），版本文件路径需要在主配置文件中指定。整包升级不需要配置源版本包。配置细节请参照本文档第3.3.5章节配置分区部分介绍。

### 3.2.3 创建配置文件

mkotapackage工具需用户配置主配置文件（必选项）和辅助配置文件，辅助配置文件为完成特殊功能，为可选配置。同时，用户可以通过mkotapackage生成配置文件，生成的配置文件可作为mkotapackage工具做包时的辅助配置文，辅助配置文件如下：

* 配置过滤器文件：升级过滤配置文件以及全量升级配置文件，请参考本文档第3.3.17章节过滤器部分；
* 生成属性文件：通过mkotapackage扫描文件夹获取属性配置文件（以分区为单位），用于做包时设置分区内文件属性，请参考文档第3.3.16章节分区文件属性部分。

### 3.2.4 生成升级包

通过mkotapackage命令和以及配置文件，生成升级包用于Update Installer安装升级。

示例如下：

|  |
| --- |
| ./mkotapackage -c partition.xml -o update.zip |

工具通过-c命令选项指定partition.xml主配置文件，通过-o命令选项指定生成升级包。

## 3.3 mkotapackage 配置文件详细说明

用户可通过主配置文件、命令行选项、辅助配置文件（可选项）配置mkotapackage工具，工具命令行选项以及主配置文件标签介绍请参考附录。

主配置文件通过命令行参数-c指定，辅助配置文件通过主配置文件中标签值指定，不直接通过命令行指定，以下介绍主配置文件和辅助配置文件使用方法。

### 3.3.1 文件规则路径

针对mkotapackage工具命令行中文件路径以及配置文件中文件路径，应遵循以下规则：

* 路径名称通过“/”分割，且不能以“/”结束；
* 文件名不包括“'”、“`”、“””、“\”等特殊字符。

### 3.3.2 mkotapackage命令行参数

mkotapackage工具命令行运行规则请参照附录，且应遵循以下规则：

* 当命令行和主配置文件配置重复功能时，优先采用命令行配置参数；
* 用户可以设置不同组合命令行参数，但不可以设置互斥的命令行参数，具体参数介绍请参考附录介绍；
* 若同时设置多个命令行选项，优先执行先设置的命令选项。若命令行选项设置多个值，优先使用先设置的值。

### 3.3.3 mkotapackage主配置文件

mkotapackage通过主配置文件（XML配置文件）设置升级包信息以及做包相关配置，设置配置文件时应注意：

* XML配置文件大小写敏感；
* XML配置文件需对齐处理；
* XML配置文件root标签<V**x**>，**x**表示版本号，mkotapackage工具版本号（工具版本号可通过命令行参数-V获取）不应低于该版本号。

主配置文件通过-c参数指定，示例如下：

|  |
| --- |
| ./mkotapackage ***–c partition.xml*** –o update.zip |

### 3.3.4 指定升级包名称

mkotapackage生成升级包名称通过命令行参数–o指定，示例如下：

|  |
| --- |
| ./mkotapackage –c partition.xml ***–o update.zip*** |

### 3.3.5 配置分区

mkotapackage生成升级包以分区为单位，分区信息需包含分区名称（必选项，不可重复）、分区类型（必选项）等参数，分区类型包括PT\_FS（有文件系统分区）和PT\_RAW（无文件系统分区）。

mkotapackage支持每次做升级包时处理单个分区，或者多个分区以及多个不同类型分区，具体配置以项目实际需求为准。

分区示例如下：

|  |
| --- |
| ……  <Partitions>  <PartitionName>PartitionName\_1</PartitionName>  <PartitionType>PT\_FS</PartitionType>  ……  </Partitions>  <Partitions>  <PartitionName>PartitionName\_2</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  ……  </Partitions>  …… |

### 3.3.6 升级包类型

通过配置xml文件中***OTAType***标签，不同分区支持不同差分类型，不同分区差分类型通过***PartitionOTAType***标签配置。当***PartitionOTAType***标签不存在时，则采用***OTAType***标签的值。

**Keys：**

|  |  |
| --- | --- |
| Value | Description |
| Incremental | 差分升级 |
| Full | 整包升级 |

差分包制作示例如下：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  <PartitionName>system</PartitionName>  ……  <Partitions>  <Partitions>  ……  <PartitionName>boot</PartitionName>  ……  <Partitions>  <v3> |

整包制作示例如下：

|  |
| --- |
| <v3>  <OTAType>full </OTAType>  <Partitions>  ……  <PartitionName>system</PartitionName>  ……  <Partitions>  <Partitions>  ……  <PartitionName>boot</PartitionName>  ……  <Partitions>  <v3> |

### 3.3.7 版本信息

通过xml配置源版本信息和目标版本信息，分别通过***SourceVersionInfo***标签和***TargetVersionInfo***标签配置，版本信息值会被附带到升级包中。

示例如下：

|  |
| --- |
| <v3>  ……  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  ……  <Partitions>  ……  <Partitions>  <v3> |

### 3.3.8 升级包压缩类型

mkotapackage支持生成压缩类型升级包和非压缩类型升级包。用户通过xml配置文件***Compress***标签配置生成升级包类型。

生成压缩类型升级包时，通过命令行参数-o指定压缩包名称（压缩类型为zip格式），生成非压缩类型升级包时，通过xml配置文件***ImageDeltaFileName***标签指定升级包名称。

**Keys：**

|  |  |
| --- | --- |
| Value | Description |
| 0 | 生成非压缩类型OTA镜像 |
| 1 | 生成压缩类型OTA包（默认） |

|  |  |
| --- | --- |
| *NOTE* | 1. 当升级包配置为非压缩类型时，需通过***ImageDeltaFileName***标签指定升级包名称。 |

示例如下：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>0</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  ……  <PartitionName>fs</PartitionName>  <PartitionType>PT\_FS</PartitionType>  ……  </Partitions>  </v3> |

### 3.3.9 配置分区信息

mkotapackage通过***Partition***标签描述分区信息，可同时配置一个升级分区或者多个分区升级，本章节下述介绍分区信息配置。

示例如下：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  <PartitionName>system</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <PartitionLocation>/dev/mmcblk0p7</PartitionLocation>  <MountPoint>/system</MountPoint>  <TargetVersion>/home/work/target/system.zip</TargetVersion>  </Partitions>  <Partitions>  <PartitionName>kernel</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  <PartitionLocation>/dev/mmcblk0p8</PartitionLocation>  <MountPoint>/kernel</MountPoint>  <TargetVersion>/home/work/target/boot.img</TargetVersion>  </Partitions>  </v3> |

### 3.3.10 分区名称

通过xml配置文件中***PartitionName***标签描述分区名称，作为每个分区的ID，该标签值为分区必选参数，且各个分区不能重复。

### 3.3.11 分区类型

通过xml配置文件中***PartitionType***标签描述分区类型，该标签值为分区必选参数。

**Keys：**

|  |  |
| --- | --- |
| Value | Description |
| PT\_FS | 分区有文件系统 |
| PT\_RAW | 分区无文件系统镜像 |

### 3.3.12 分区文件系统类型

通过xml配置文件中***FileSystem***标签描述分区文件系统类型。

|  |  |
| --- | --- |
| *NOTE* | 1.只支持日志型文件系统，该选项在分区类型是PT\_FS和PT\_RAW时都有效 |

目前支持分区文件系统类型如下：

**Keys**：

|  |  |
| --- | --- |
| Value | Description |
| EXT4  EXT2 | Ext4类型文件系统  Eex2类型文件系统 |
| UBIFS | UBIFS类型文件系统 |
| DOS | DOS类型文件系统 |
| QNX4  QNX6  MTD  EMMC  NOR | QNX4类型文件系统  QNX6类型文件系统  MTD裸设备  EMMC设备  NOR裸设备 |

### 3.3.13 分区设备节点号

针对PT\_RAW分区，通过xml配置文件中***PartitionLocation***标签配置分区在系统中的设备节点号。

### 3.3.14 分区挂载点

通过xml配置文件中***MountPoint***标签配置分区的挂载目录。

|  |  |
| --- | --- |
| *NOTE* | 1. 基于镜像分区的升级可忽略该设置项。 2. 挂载点请使用绝对路径。 3. 挂载点支持一级目录或多级目录点的挂载。 |

### 3.3.15 分区文件路径

通过xml配置文件中***SourceVersion***标签和***TargetVersion***设置分区的源版本文件路径和目标版本的文件路径。

如果分区类型为PT\_RAW，则配置路径为单一的分区镜像文件。

如果分区类型为PT\_FS，则要求对该分区的源版本文件先进行打包再压缩，压缩格式支持zip、tar.gz、tgz，且保证解压后为根路径。

|  |  |
| --- | --- |
| *NOTE* | 1. 基于镜像的分区，给出镜像的绝对路径（包含镜像文件名） 2. 如果是升级类型是full，可不用配置SourceVersion字段 |

示例如下：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  <SourceVersion>***/home/source\_bak/source\_system.zip***</SourceVersion>  <TargetVersion>***/home/target\_bak/target\_system.zip***</TargetVersion>  ……  </Partitions>  </v3> |

### 3.3.16 配置分区文件属性

针对PT\_FS类型分区，需用户指定分区文件属性，mkotapackage工具支持指定属性文件方式和自动扫描分区文件属性方式。

|  |  |
| --- | --- |
| *NOTE* | 1. 分区类型PF\_FS有效；  2. 建议使用指定属性文件方式，若使用工具扫描文件属性则需用户保证扫描目录文件的正确性。 |

#### 3.3.16.1 指定属性文件

该方式需用户先手动获取属性文件，然后通过***SourceAttributesFile***标签和***TargetAttributesFile***标签指定属性文件。

1. 用户根据升级的源版本和目标版本获取文件属性，文件属性文件格式以行为单位，描述文件属性；

|  |
| --- |
| "<file name>" <type><attribute string size><attribute string> |

其中（属性为必须项）：

* file name：文件名，路径以指定path为相对路径，用双引号区分；
* type：文件类型，目前支持有Regular普通文件、SymLink链接文件和Directoy目录文件；
* attribute string size：文件属性描述字符串长度；
* attribute string：文件属性，包括：文件权限、用户ID和用户组ID。

示例如下：

|  |
| --- |
| "/boot/diff\_file\_4" Regular 14 0644:1000:1000 |

2. 用户将目标版本和源版本文件属性结果保存到文件中（如有需要用户可根据实际情况修改文件属性），如保存为source\_attributes.txt文件和target\_attributes.txt文件。

此外，通过mkotapackage工具可帮助用户扫描目录生成属性文件，通过mkotapackage–s path命令扫描目录（此方式需用户保证被扫描目录的文件正确性），如下：

|  |
| --- |
| ./mkotapackage –s /home/source/boot/  ./mkotapackage –s /home/target/boot/ |

将上述命令输出分别保存为source\_atrributes.txt和target\_attributes.txt属性文件即可；

3. 在主配置文件中，通过***SourceAttributesFile***标签和***TargetAttributesFile***标签指定属性文件，如下示例：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  < SourceAttributesFile >***. /source\_attributes.txt***</ SourceAttributesFile>  < TargetAttributesFile >.***/target\_attributes.txt***</ TargetAttributesFile >  ……  </Partitions>  </v3> |

#### 3.3.16.2 指定扫描目录路径

mkotapackage支持自动扫描文件属性，当没有属性文件时，用户可通过xml配置文件中***SourceAttributesFromFolder***和***TargetAttributesFromFolder***标签配置源文件和目标文件所在的文件夹，用于扫描该分区中所有的文件属性（此模式需用户保证配置的文件夹下文件的正确性）。

示例如下：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  < SourceAttributesFromFolder >***/home /source\_system***</ SourceAttributesFromFolder >  < TargetAttributesFromFolder >***/home /target\_system***</ TargetAttributesFromFolder >  ……  </Partitions>  </v3> |

### 3.3.17 配置文件过滤器

通过SourceExclude标签和TargetExclude标签可配置过滤器脚本路径。配置脚本支持通配符和文件夹，并且每一项规则以换行符结束。

考虑到一些升级分区并非是只读的，设备中软件版本会随着系统运行部分文件会发生变化。为了支持这些发生改变的文件升级，mkotapackage内部设计了一个过滤器，过滤器在做包之前，根据***SourceExcludeFilter***/***TargeExcludeFilter***指定的过滤配置文件内的过滤规则，将用户输入的***SourceVersion***/***TargetVersion***版本文件内符合过滤规则的文件先剔除掉，然后在重新打包压缩生成源和目标文件送去进行做包。

|  |  |
| --- | --- |
| *NOTE* | 1. 支持标准的GLOB操作。 2. 仅对差分类型有效。 |

过滤器处理原理如下图所示。



如下实例：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  < SourceExcludeFilter >***./source\_exclude.txt***</ SourceExcludeFilter>  < TargeExcludeFilter>***./target\_exclude.txt***</ TargeExcludeFilter>  ……  </Partitions>  </v3> |

用户需在source\_exclude.txt和target\_exclude.txt文件中指定需要过滤的文件，也支持过滤目录，工具在做包时会自动过滤指定的文件。

### 3.3.18 强制覆盖过滤器

可通过***VerbatimFilter***配置verbatim过滤器配置文件路径，该配置文件内所配置的所有文件将被全量带进升级包。

|  |  |
| --- | --- |
| *NOTE* | 1. 支持标准的GLOB操作。 2. 仅对差分类型有效。 |

如下实例：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  < VerbatimFilter>./verbatim.txt</ VerbatimFilter >  ……  </Partitions>  </v3> |

用户需在verbatim.txt文件中指定需要全量升级的文件，工具在做包时会过滤这些文件，并将这些文件全量带入升级包中。

### 3.3.19 Not-In-Place方式升级

默认情况下mkotapackage做出的升级包支持In-Place方式升级（源位置和目标安装位置相同），同时mkotapackage做出的包也支持Not-In-Place方式升级（源位置与目标安装位置不同），以Not-In-Place方式升级时，做包时需做如下设置：

1. PT\_FS类型分区

针对PT\_FS类型分区，需配置分区源位置与目标安装位置，通过主配置文件中：

* ***PartitionInputDir***标签设置源安装位置，若不设置该标签则mkotapackage默认采用***MountPoint***标签的参数值；
* ***PartitionOutputDir***标签设置目标安装目录，若不设置该标签值则默认为In-Place方式升级包。

2. PT\_RAW类型分区

针对PT\_RAW类型分区，需配置分区源位置与目标安装位置，通过主配置文件中：

* ***PartitionInputLocation***标签设置源安装位置；
* ***PartitionOutputLocation***标签设置目标安装目录，若不设置该标签值则默认为In-Place方式升级包。

### 3.3.20 升级只读文件系统

mkotapackage支持以镜像升级方式升级只读文件系统，用户需提供该只读文件系统镜像，配置如下：

* 指定分区类型为PT\_RAW类型；
* 用户生成源版本和目标版本分区镜像文件，通过***SourceVersion***和***TargetVersion***标签指定路径；

示例如下：

|  |
| --- |
| <v3>  <Partitions>  <PartitionName>system</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  ……  <SourceVersion>/home/ source/system.img</SourceVersion>  <TargetVersion>/home/target/system.img</TargetVersion>  ……  </Partitions>  </v3> |

### 3.3.21 配置分块升级

针对PT\_RAW类型分区，mkotapackage工具支持分块升级功能，用户需手动配置分块信息，通过BlockSegment、BlockOffset、BlockSize等标签设置分块属性。

|  |  |
| --- | --- |
| *NOTE* | 1. BlockSegment跟BlockOffset和BlockSize两个属性标签配合使用。 2. 仅分区为PT\_RAW类型有效。 3. 仅差分升级模式有效，且目前仅支持bsdiff算法。 |

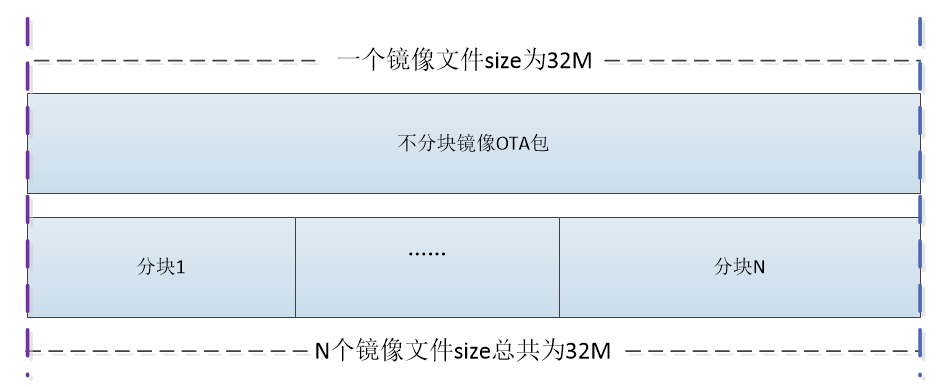
* BlockSegment：用于描述分块信息的标识符。
* BlockOffset：用于设置分块起始地址，第N块的起始地址= 前1块（第N-1块）的起始地址+ 前1块（第N-1块）的大小。

|  |  |
| --- | --- |
| *NOTE* | 1. 偏移地址的数值是十进制(单位：B)。 2. 跟BlockSize配套使用。 |

BlockSize：用来设置分块的大小，-1表示剩余分区文件size。

|  |  |
| --- | --- |
| *NOTE* | 1. 分块大小的数值是十进制(单位：B)。 2. 跟BlockOffset配套使用。 3. 几个块可以设置成不同的大小。 |

如制作升级包时未设置分块功能，1个分区制作出的升级包镜像文件为1个（如size为32M），在制作升级包时将分区设置分为n块，则会对应生成n个镜像文件，n个文件总计大小为32M，如下图：



设置分块功能，假设平均设置为4块，每块大小为8MB，则配置如下：

每块的BlockSize数在配置表中，应配置数为：8 \* 1024 \* 1024 = 8388608

每块的起始地址= 前一块的起始地址（BlockOffset）+ 前一块的偏移量（BlockSize）

分块2的起始地址= 0（分块1的起始地址）+ 8388606（分块1的偏移量） = 8388608

分块3的起始地址= 8388608（分块2的起始地址）+ 8388608（分块2的偏移量）= 16777216

分块4的起始地址= 16777216（分块3的起始地址+ 8388608（分块3的偏移量）=25165824

配置文件中对应如下设置：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  ……  <BlockSegment>  <BlockOffset>0</BlockOffset> //分块1起始地址  <BlockSize>8388608</BlockSize> //分块1大小  </BlockSegment>  <BlockSegment>  <BlockOffset>8388608</BlockOffset> //分块2起始地址  <BlockSize>8388608</BlockSize> //分块2大小  </BlockSegment>  <BlockSegment>  <BlockOffset>16777216</BlockOffset> //分块3起始地址  <BlockSize>8388608</BlockSize> //分块3大小  </BlockSegment>  <BlockSegment>  <BlockOffset>16777216</BlockOffset> //分块4起始地址  <BlockSize>8388608</BlockSize> //分块4大小  </BlockSegment>  ……  </Partitions>  </v3> |

### 3.3.22 SElinux支持

SELinux是一种基于域-类型模型（domain-type）的[强制访问控制](https://baike.baidu.com/item/%E5%BC%BA%E5%88%B6%E8%AE%BF%E9%97%AE%E6%8E%A7%E5%88%B6/3617989)（MAC）[安全系统](https://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E7%B3%BB%E7%BB%9F/3131501)，本文档不做详细介绍。mkotapackage支持SElinux功能（需特定版本支持），如果分区开启了SElinux功能，需要设置SElinuxSupport为true，默认为false。

### 3.3.23 内存优先支持

通过xml配置文件中***updater\_memoryfirst***标签配置是否使用内存优先方式进行升级。

使用内存优先方式占用内存资源较多，但升级还原速度较快；不使用内存优先则占用内存较少，但速度较慢。

当内存较为充裕，满足《Installer4.0集成操作手册-Installer》中“RAM空间需求”章节速度优先的ram空间要求时，建议使用内存优先方式，否则使用省内存方式。

|  |  |
| --- | --- |
| *NOTE* | 1. 设置值为0（默认值），则代表升级包对此不做要求，使用设备端原预设的内存优先/省内存方式。  2. 设置值为1，则代表升级包通知Installer升级程序使用内存优先方式，设备端原预设的内存优先/省内存方式无效。  3. 设置值为2，则代表升级包通知Installer升级程序使用省内存方式，设备端原预设的内存优先/省内存方式无效。 |

### 3.3.24 优先调用本地升级程序

通过xml配置文件中***PolicyUpdaterUsefunction***标签设置是否使用本地installer内置的升级脚本执行程序，代替升级包中的updater。

|  |  |
| --- | --- |
| *NOTE* | 1. 设置值为0（默认值），则代表升级包对此不做要求，使用设备端原预设的updater\_usefunction方式；  2. 设置值为1，则代表升级包通知升级程序在升级时调用动态库中的函数接口。设备端原预设的updater\_usefunction方式无效；  3. 设置值为2，则代表升级包通知升级程序优先使用设备端本地的升级程序，而不调用动态库中的函数接口，设备端原预设的updater\_usefunction方式无效；  备注：当做包时使用-u参数向升级包中添加updater-binary文件时，在设备端还原升级时优先使用升级包中的updater-binary。 |

### 3.3.25 设置可运行的最大并行线程数

通过xml配置文件中***PolicyThreadCount***标签设置可运行的最大并行线程数，做并发升级时使用。

|  |  |
| --- | --- |
| *NOTE* | 1. 设置值为0，则代表升级包对此不做要求，使用设备端原预设的最大并行线程数。  2. 设置值为1-100，则代表升级包通知升级程序使用的最大并行线程数值。 |

### 3.3.26 设置预升级功能

通过xml配置文件中***PolicyPreUpdate***标签配置是否使用预升级功能。

使用预升级功能对升级包执行升级，但不进行实际的写入操作。用于验证整个升级流程是否有错误发生。

|  |  |
| --- | --- |
| *NOTE* | 1. 设置值为0，则代表升级包对此不做要求，使用设备端原预设升级模式。  2. 设置值为1，则代表升级包通知升级程序使用预升级功能。  3. 设置值为2，则代表升级包通知升级程序不使用预升级功能。 |

### 3.3.27 定时器设置

mkotapackage生成升级包时，可设置超时时间。当到达超时时间时mkotapackage工具会退出做包过程。

mkotapackage默认设置超时时间为5min，且用户可通过mkotapackage–T命令手动设置超时时间。

### 3.3.28 日志统计系统

mkotapackage生成升级包过程会生成log文件，默认生成log文件路径为当前程序执行路径下log文件夹。

### 3.3.29 配置镜像文件自动挂载

针对PT\_FS类型的文件系统分区，用户可输入一个标准的image文件、Ubifs镜像文件或者Sparse 镜像文件，mkotapackage会自动进行挂载、打包压缩、扫描属性等操作。

Mkotapackage工具在进行自动挂载、打包、扫描等动作时，需用户通过***SourceVersionImageType*** /***TargetVersionImageType***标签描述镜像文件类型，目前支持镜像类型如下表：

|  |  |
| --- | --- |
| **镜像类型** | **描述** |
| STD-IMG | 标准的image文件 |
| SPARSE-IMG | sparse image文件 |
| UBIFS-IMG | ubifs 镜像文件 |

**备注：**

|  |  |
| --- | --- |
| *NOTE* | 1.使用此项配置后，工具会自动挂载、打包并扫描属性SourceAttributesFromFolder和TargetAttributesFromFolder不需要再使用；  2.UBIFS-IMG需要系统支持nandsim模块以及mtd-utils工具集支持；  **3.此功能在工具中增加对上述不同镜像文件的自动挂载、打包和扫描等动作，且工具中集成的为标准镜像操作，实际项目中处理镜像时可能会出现挂载失败或者其他异常错误，则此时建议用户根据实际镜像文件手动挂载、按分区打包使用SourceAttributesFromFolder和TargetAttributesFromFolder标签进行做包。** |

### 3.3.30 配置整版本校验功能

默认情况下，Installer4.0在升级过程中仅会校验参与差分升级的文件（升级前会校验哈希值，升级后也会进行哈希值的校验操作）。同时Installer4.0也支持校验包括不参与差分升级的所有文件，此时用户需要在做包时候配置***PolicyPackageFullSHACheck***标签，该配置表示在升级时，会当前系统中所有的文件（包括不参与差分升级部分文件）。同时做包时可配置***PolicyUpdaterWithSHACheck***标签，配置在校验文件时出错是否会退出升级流程。

|  |  |
| --- | --- |
| *NOTE* | 1. PolicyPackageFullSHACheck默认参数为0，其中：  0：表示升级包对此不做要求，使用设备端默认配置（设备端还原时默认配置一般为2）；  1：表示执行所有文件校验；  2：表示不进行 所有文件校验。  2. PolicyUpdaterWithSHACheck默认参数为0，其中：  0：表示升级包对此不做要求，使用设备端默认配置（设备端还原时默认配置一般为2）；  1：表示校验该部分文件失败时，会退出当前的升级流程；  2：表示校验该部分文件失败时不会影响升级流程，即不退出升级。 |

## 3.4 主配置文件实例

以下为xml配置实例，用户配置使用时应以实际项目需求为准。

### 3.4.1 In-Place方式文件系统分区配置文件

示例如下，为生成一个压缩类型差分包，用于In-Place方式升级。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>1</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>fs</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <FileSystem>EXT4</FileSystem>  <MountPoint>/fs\_test</MountPoint>  <SourceAttributesFromFolder>/home/source\_bak/system</SourceAttributesFromFolder>  <TargetAttributesFromFolder>/home/target\_bak/system</TargetAttributesFromFolder>  <SourceVersion>/home/source\_bak/source\_system.zip</SourceVersion>  <TargetVersion>/home/target\_bak/target\_system.zip</TargetVersion>  </Partitions>  <Partitions>  <PartitionName>app</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <FileSystem>EXT4</FileSystem>  <MountPoint>/app\_test</MountPoint>  <SourceAttributesFromFolder>/home/source\_bak/app</SourceAttributesFromFolder>  <TargetAttributesFromFolder>/home/target\_bak/app</TargetAttributesFromFolder>  <SourceVersion>/home/source\_bak/source\_app.zip</SourceVersion>  <TargetVersion>/home/target\_bak/target\_app.zip</TargetVersion>  </Partitions>  </v3> |

### 3.4.2 Not-In-Place方式文件系统分区配置文件

针对Not-In-Place方式升级，生成升级包时需通过***PartitionInputDir***标签指定源目录，***PartitionOutputDir***标签指定目标安装目录（不配置该参数则默认为In-Place方式升级），示例如下，生成一个压缩类型差分包，用于Not-In-Place方式升级。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>1</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>system</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <FileSystem>EXT4</FileSystem>  <MountPoint>/fs\_test</MountPoint>  <PartitionInputDir>/ fs\_test </PartitionInputDir>  <PartitionOutputDir>/system\_new</PartitionOutputDir>  <SourceAttributesFromFolder>/home/source\_bak/system</SourceAttributesFromFolder>  <TargetAttributesFromFolder>/home/target\_bak/system</TargetAttributesFromFolder>  <SourceVersion>/home/source\_bak/source\_system.zip</SourceVersion>  <TargetVersion>/home/target\_bak/target\_system.zip</TargetVersion>  </Partitions>  </v3> |

### 3.4.3 非压缩类型升级包配置文件

非压缩类型升级包需配置***Compress***标签值为0，同时通过***ImageDeltaFileName***指定升级包名称。示例如下，为生成一个非压缩类型差分包，用于In-Place方式升级，生成升级包名称为update.bin。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>0</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>fs</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <FileSystem>EXT4</FileSystem>  <MountPoint>/fs\_test</MountPoint>  <SourceAttributesFromFolder>/home/source\_bak/system</SourceAttributesFromFolder>  <TargetAttributesFromFolder>/home/target\_bak/system</TargetAttributesFromFolder>  <SourceVersion>/home/source\_bak/source\_system.zip</SourceVersion>  <TargetVersion>/home/target\_bak/target\_system.zip</TargetVersion>  </Partitions>  </v3> |

### 3.4.4 镜像分区升级配置文件

示例如下，针对PT\_RAW类型分区，生成一个压缩类型差分包，用于In-Place方式升级。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>1</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>kernel</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  <FileSystem>EMMC</FileSystem>  <PartitionLocation>/dev/mmcblk0p7</PartitionLocation>  <SourceVersion>/home/work/source/zImage</SourceVersion>  <TargetVersion>/home/work/target/zImage</TargetVersion>  </Partitions>  </v3> |

### 3.4.5 镜像分区分块升级配置文件

示例如下，针对PT\_RAW类型分区，生成一个压缩类型差分包，用于In-Place方式升级，且配置分块升级。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>1</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>kernel</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  <FileSystem>EMMC</FileSystem>  <PartitionLocation>/dev/mmcblk0p7</PartitionLocation>  <MountPoint>/boot</MountPoint>  <BlockSegment>  <BlockOffset>0</BlockOffset> //分块1起始地址  <BlockSize>8388608</BlockSize> //分块1大小  </BlockSegment>  <BlockSegment>  <BlockOffset>8388608</BlockOffset> //分块2起始地址  <BlockSize>8388608</BlockSize> //分块2大小  </BlockSegment>  <BlockSegment>  <BlockOffset>16777216</BlockOffset> //分块3起始地址  <BlockSize>8388608</BlockSize> //分块3大小  </BlockSegment>  <BlockSegment>  <BlockOffset>16777216</BlockOffset> //分块4起始地址  <BlockSize>-1</BlockSize> //分块4大小  </BlockSegment>  <SourceVersion>/home/work/source/zImage</SourceVersion>  <TargetVersion>/home/work/target/zImage</TargetVersion>  </Partitions>  </v3> |

### 3.4.6 只读文件系统分区升级配置文件

只读文件系统分区升级以镜像方式升级，用户需提供该只读文件系统镜像。示例如下，针对只读的PT\_FS分区，以镜像方式升级。

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <SourceVersionInfo>v1.0</SourceVersionInfo>  <TargetVersionInfo>v2.0</TargetVersionInfo>  <Compress>1</Compress>  <ImageDeltaFileName>update.bin</ImageDeltaFileName>  <Partitions>  <PartitionName>boot</PartitionName>  <PartitionType>PT\_RAW</PartitionType>  <FileSystem>EMMC</FileSystem>  <PartitionLocation>/dev/mmcblk0p7</PartitionLocation>  <MountPoint>/boot</MountPoint>  <SourceVersion>/home/work/source/boot.img</SourceVersion>  <TargetVersion>/home/work/target/boot.img</TargetVersion>  </Partitions>  </v3> |

### 3.4.7 镜像自动挂载分区升级配置文件

针对标准镜像、ubifs镜像、sparse镜像文件，做包工具支持自动挂载、打包、扫描属性操作，配置文件如下示例：

|  |
| --- |
| <v3>  <OTAType>incremental</OTAType>  <Partitions>  <PartitionName>app</PartitionName>  <PartitionType>PT\_FS</PartitionType>  <FileSystem>ext4</FileSystem>  <MountPoint>/emmc</MountPoint>  <SourceVersion>./source/source.ext4</SourceVersion>  <TargetVersion>./target /target.ext4</TargetVersion>  <SourceVersionImageType>STD-IMG</SourceVersionImageType>  <TargetVersionImageType>STD-IMG</TargetVersionImageType>  </Partitions>  </v3> |

# 附录

下面对mkotapackage工具的命令行参数及配置文件参数附加说明。

## 4.1 mkotapackage命令行参数

* mkotapackage -h

**Description**

帮助参数。列出mkotapackage所有参数已经对应功能说明。如下：

|  |
| --- |
| usage: mkotapackage [-h] [--version] [--updater <updater>]  [--diff\_timeout <second>] [--full\_time <second>]  [--patch\_threshold <percentage>] [--out\_fstab <file>]  [--out\_excludes <file>] [--extra\_data <file> [<file> ...]]  [--configure <configure\_file>] [--show\_configure\_key]  [--output <file>] [--scan\_attributes <path>]  [--errno\_log\_file <path>] [--output\_log\_file <path>]  [--statistics\_file <path>]  [--incremental <file> <file> <file>] [--get\_os\_version]  [--get\_start\_address] [--package\_path <file>] [--verify]  [--ramsize <K unit>]  This is a command line tool to build an OTA package  optional arguments:  -h, --help show this help message and exit  --version, -V output the version number and exit  --updater <updater>, -u <updater>  the program binary for updating, add it to OTA package  --diff\_timeout <second>, -T <second>  the max time(second) to compute delta between files,  the default 300 seconds is used if not set  --full\_time <second>, -f <second>  set the actually time that full upgrade finished  needed, the default 30 seconds is used if not set  --patch\_threshold <percentage>, -S <percentage>  set to the patch threshold and the default is 0.95,  cover the target file directly if the patch file more  than patch\_threshold of the target file and the  process will not check the source file SHA value  --out\_fstab <file>, -t <file>  output mount volume table for debug  --out\_excludes <file>, -x <file>  output the exclude file information for debug  --extra\_data <file> [<file> ...], -e <file> [<file> ...]  add the extra file into the OTA package archive  pointed by the option -o  --configure <configure\_file>, -c <configure\_file>  the base target package configure file, the xml form  is only supported currently  --show\_configure\_key, -C  show all of available configuration key in the xml  configure file  --output <file>, -o <file>  the output OTA package archive  --scan\_attributes <path>, -s <path>  scan the attributes information of specified directory  --errno\_log\_file <path>, -errno\_log <path>  save the error code status of specified file  --output\_log\_file <path>, -output\_log <path>  save the output log to specified file  --statistics\_file <path>, -statistics <path>  save the statistics file to specified file  --incremental <file> <file> <file>, -i <file> <file> <file>  make an OTA(incremental) package by two full package  --get\_os\_version get the OTA package version  --get\_start\_address get the OTA package start address  --package\_path <file>, -p <file>  the OTA package path  --verify, -v verify the OTA package  --ramsize <K unit>, -ramsize <K unit>  Pre-alloced RAM space for update, default 1024000 |

mkotapackage-V

**Description**

该参数打印mkotapackage的版本号、生成时间。

mkotapackage -u

**Description**

该参数用来指定输入升级程序updater文件路径。updater包含了升级核心算法，由OTA服务商根据目标平台交叉编译产生。

|  |  |
| --- | --- |
| *NOTE* | 1. updater文件由OTA 服务商提供。 |

mkotapackage -T

**Description**

该参数用来设置单个文件差分运算时超时退出时间

|  |  |
| --- | --- |
| *NOTE* | 1. 单位是：秒。 2. 如未设置，默认是300（5分钟）。 |

mkotapackage –full\_time

**Description**

该参数用来设置整包升级的时间(该时间由用户指定，用于较准确的模拟整包升级的进度)。

|  |  |
| --- | --- |
| *NOTE* | 1. 单位是：秒。 2. 如未设置，默认是30，改时间可先设置30后续根据实际测试时间为准。 |

Mkotapackage -t

**Description**

指定输出分区表文件的路径。该分区表文件是根据xml文件配置的分区消息，将每个分区按照固定格式保存到该文件中。该分区表文件可提供给UA模块使用。文件格式如下：

#PartitionLocatition #mountpoint #filesystem #mountflags #mountoptions

/dev/sda1 /system ext4 rw,barrier=1 wait,sync

|  |  |
| --- | --- |
| *NOTE* | 1.该文件只记录分区类型为PT\_FS的分区信息。  2.不包含cache分区信息。 |

mkotapackage -x

**Description**

指定过滤器匹配日志文件。对于一些非只读的文件系统的升级，可能会遇到一些系统文件发生变化，mkotapackage内部支持一个过滤器支持系统中已经发生变化文件升级。该选项用于保存过滤器在运行过程匹配到的文件列表。

|  |  |
| --- | --- |
| *NOTE* | 1. 过滤器配置文件通过xml中的SourceExcludeFilter/TargetExcludeFilter配合使用  2.只读分区无需使用 |

mkotapackage-e

**Description**

该参数用来增加额外文件至升级包中。

增加进来的额外文件，Installer4..0模块提供对应的单独函数，用来提取这些额外文件到用户指定的路径，以便用户根据需要，用例实现自定义功能。Installer4.0模块提供的升级函数在对设备升级时，不包括这些添加进来的额外文件。

|  |  |
| --- | --- |
| *NOTE* | 1.一次调用只能增加一个文件 |

mkotapackage -c

**Description**

该参数用来设置mkoatpackage xml配置文件的路径。Xml配置文件介绍见下一章。

|  |  |
| --- | --- |
| *NOTE* | 1.必选项。 |

mkotapackage -C

**Description**

该参数可显示xml配置文件中可使用的元素。

mkotapackage -o

**Description**

该参数用来设置mkotapackage输出的升级包文件路径。

mkotapackage -s

**Description**

该参数指定源或目标文件夹路径，并扫描打印文件属性信息。

mkotapackage -S

**Description**

该参数设置差分文件的最大差分比率。当差分文件和目标文件的比率超过设置的最大的查分比之后，程序内部会优化选择直接装载目标文件；默认是0.95，设置为0,为不优化。

mkotapackage -l

**Description**

该参数设置设置错误码路径，默认路径为做包工具当前执行路径。

mkotapackage-i

**Description**

该参数通过基础版本包进行差分操作，该参数需输入3个参数：基础版本号v1、基础版本V2、以及差分输出目标包。

如果基本版本V2中存在通过-e添加的用户数据时，则同时提取V2中用户数据至目标包（V1包中用户数据忽略）。

mkotapackage –-get\_os\_version

**Description**

该参数设置为获取基础包版本号，需与-p参数配合使用。

mkotapackage-p

**Description**

该参数指定基础包路径，与—get\_os\_version配合使用。

## 4.2 mkotapackage配置文件参数说明

为了满足通用户针对不同情况的需求，用户可通过设置OTA包制作工具的xml配置，来满足不同情景需求。当前版本配置文件支持的属性项list如下：

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| OTAType | OTA包的类型 |
| SourceVersionInfo | 源版本描述信息 |
| TargetVersionInfo | 目标版本描述信息 |
| Compres | OTA包压缩类型 |
| ImageDeltaFileName | 生成非压缩包名称 |
| Partitions | 分区的分割符标签 |
| PartitionType | 分区类型 |
| PartitionOTAType | 分区差分类型 |
| FileSystem | 分区文件系统类型 |
| PartitionInputDir | PT\_FS分区源安装目录 |
| PartitionOutputDir | PT\_FS分区目标安装目录 |
| PartitionInputLocation | PT\_RAW分区源安装设备节点号 |
| PartitionOutputLocation | PT\_RAW分区目标安装设备节点号 |
| PartitionName | 分区名称 |
| PartitionLocation | 分区设备号 |
| MountPoint | 分区的挂载路径 |
| SourceVersion | Source（压缩包或镜像）的路径 |
| TargetVersion | Target（压缩包或镜像）的路径 |
| SourceAttributesFromFolder | 待扫描源版版本文件属性的分区文件路径 |
| TargetAttributesFromFolder | 待扫描目标版本文件属性的分区文件路径 |
| SourceExcludeFilter | 源版本的过滤器配置文件 |
| TargetExcludeFilter | 目标版本的过滤器配置文件 |
| VerbatimFilter | 升级包过滤器配置文件 |
| BlockSegment | 分区分块标识符 |
| BlockOffset | 分块的起始地址 |
| BlockSize | 分块的大小 |
| SourceAttributesFile | 源文件的属性配置文件 |
| TargetAttributesFile | 目标文件的属性配置文件 |
| SElinuxSupport | SElinux属性支持 |
| PolicyUpdaterMemoryFirst | 内存优先模式支持 |
| PolicyUpdaterUsefunction | 使用installer自带的升级脚本解析程序 |
| PolicyThreadCount | 设置可运行的最大并行线程数 |
| PolicyPreUpdate | 设置预升级功能 |
| PolicyPackageFullSHACheck | 设置是否执行整版本校验功能 |
| PolicyUpdaterWithSHACheck | 设置整版本校验时是否影响升级流程 |