

KMZ任务文件说明

一、KMZ文件说明

`waypoints_name.kmz`

| - `wpmz`

| - `res`

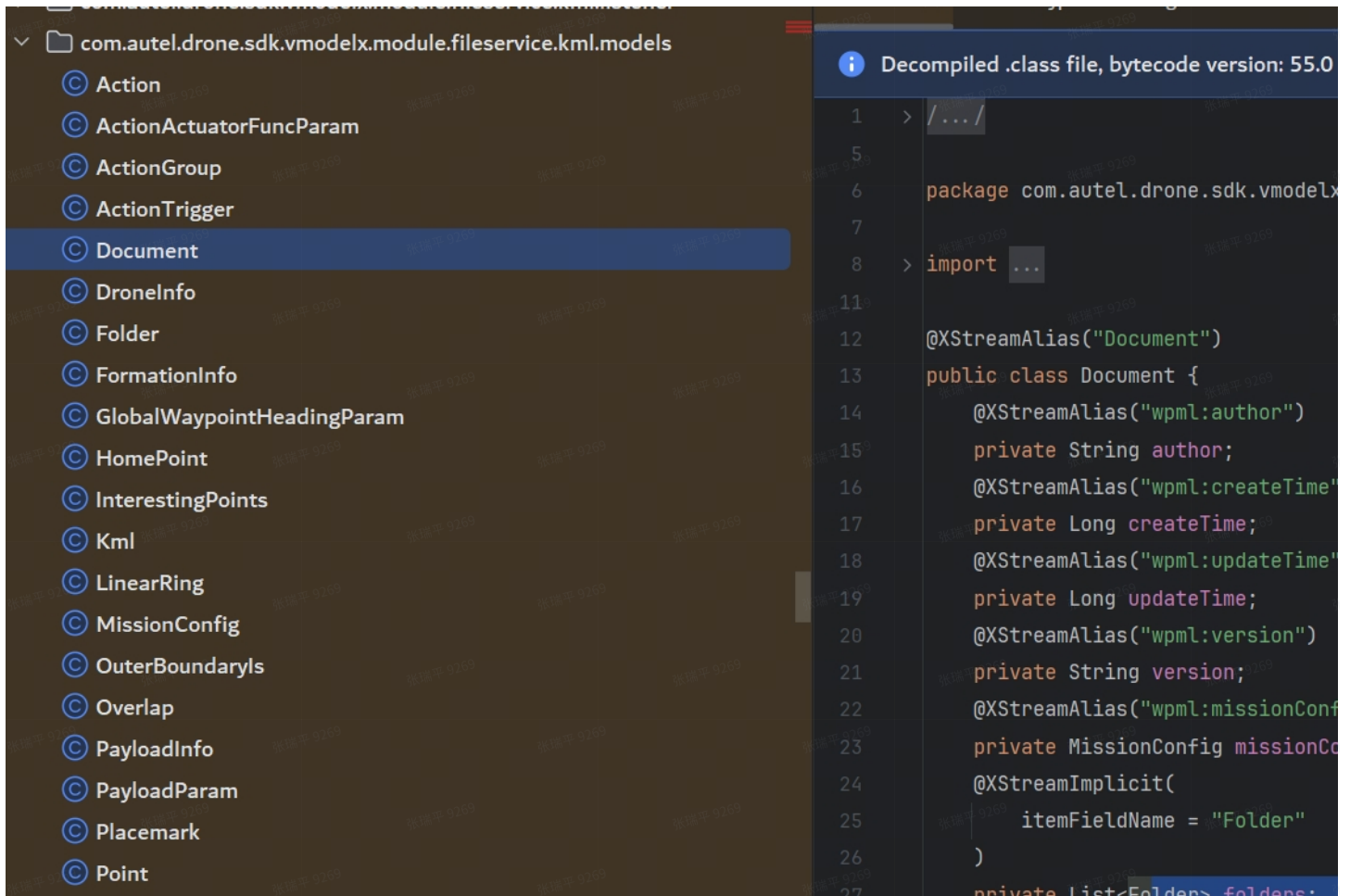
| - `template.kml`

| - `waylines.wpm1`

WPML 是 **WayPoint Markup Language** 的缩写，即航线文件格式标准。WPML 航线文件格式标准基于 KML（Keyhole Markup Language）的定义进行扩展。WPML 航线文件遵循 KMZ 归档要求，所有航线文件以 “.kmz” 后缀结尾。WPML 航线文件格式标准作为航线数字资产的载体。

1. 本质是一个zip压缩文件，kml和wpm1均为xml文档格式
2. `template.kml` 文件被称为 “模板文件”，方便用户编辑规划
3. `waylines.wpm1` 文件被称为 “执行文件”
4. `res` 资源文件夹包含了航线所需的辅助资源，如在开始精准复拍前，预先准备的参考目标物照片等

二、MSDK对KMZ支持



1.定义了KML对象模型



2.MissionConfig为全局配置



三、KML对象到xml文档

1.KML 对象填充数据后，通过XStream转换成 xml文本

```
1 val xmlUtils = XmlUtils<Kml>("Autel")
2 val wpmlPackager = WpmlPackager()
3 val kmlBean = wpmlPackager.pack(flightModel, msnInfoUsr)
4 val xmlString = xmlUtils.objectToXml(kmlBean )
```

2.依次生成 `template.kml` `waylines.wpml` ，设定好目录压缩成zip，扩展名为kmz

四、KMZ任务执行控制

1.上传kmz文件到飞机，然后执行kmz任务

1 任务上传：

```

2
3     val guid = System.currentTimeMillis() / 1000
4     missionManager =
        DeviceManager.getDeviceManager().getFirstDroneDevice()?.getWayPointMissionManager()
5
6     missionManager?.uploadKmlMissionFile(kmzFilePath, guid.toInt(), object:
        CommonCallbacks.CompletionCallbackWithProgressAndParam<Long> {
7         override fun onProgressUpdate(progress: Double) {}
8         override fun onSuccess(guid: Long?) { }
9         override fun onFailure(error: IAutelCode, msg: String?) {}
10    })

```

```

1 任务控制：(注意参数为MissionKmlGUIDBean)
2 missionManager?.startMission(MissionKmlGUIDBean(it.toInt()),
3     object : CommonCallbacks.CompletionCallbackWithParam<Void> {
4         override fun onFailure(error: IAutelCode, msg: String?) {}
5         override fun onSuccess(t: Void?) {}
6     })
7
8
9 val isKml = true
10 missionManager?.pauseMission(object:
    CommonCallbacks.CompletionCallbackWithParam<Void>{
11     override fun onFailure(error: IAutelCode, msg: String?) {}
12     override fun onSuccess(t: Void?) {}
13 }, isKml)
14
15 ...

```

五、参考

1.kmz详解: https://doc.autelrobotics.com/cloud_api/cn/60/00/10

2.演示 MSDK2.0-MissionDemo