# KMZ任务文件说明

一、KMZ文件说明

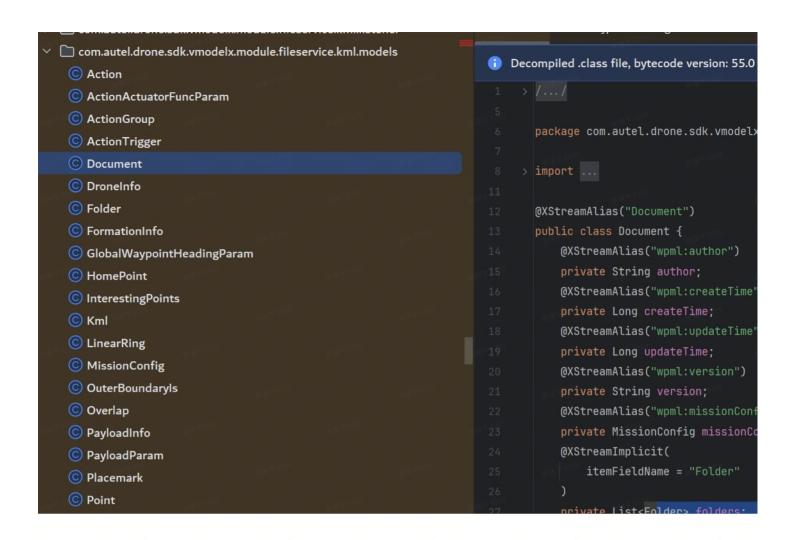
waypoints\_name.kmz |- wpmz

- res
- template.kml
- waylines.wpml

WPML 是 **W**ay**P**oint **M**arkup **L**anguage 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。

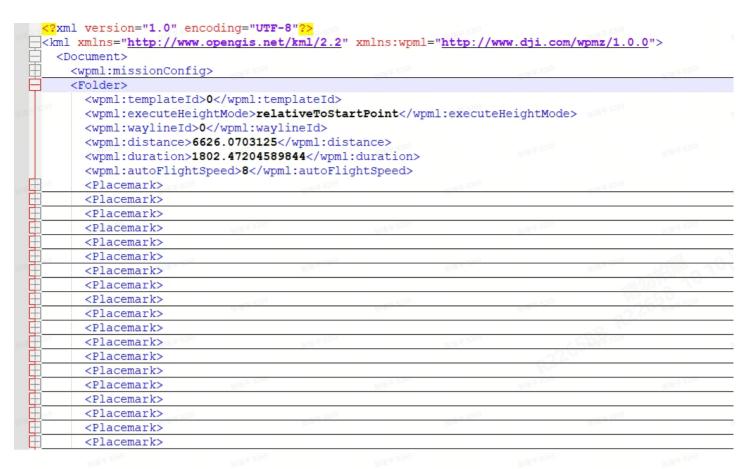
- 1.本质是一个zip压缩文件,kml和wpml均为xml文档格式
- 2. template.kml 文件被称为 "模板文件" ,方便用户编辑规划
- 3. waylines.wpml 文件被称为 "执行文件"
- 4. res 资源文件夹包含了航线所需的辅助资源,如在开始精准复拍前,预先准备的参考目标物照片等

## 二、MSDK对KMZ支持



#### 1.定义了KML对象模型

## 2.MissionConfig为全局配置



#### 3.Placemark代表一个航点

Point为点的经纬度坐标

ActionGrop为动作集合

Action为具体一个动作,如拍照

```
<Folder>
 <wpml:templateId>0</wpml:templateId>
  <wpml:executeHeightMode>relativeToStartPoint</wpml:executeHeightMode>
 <wpml:waylineId>0</wpml:waylineId>
 <wpml:distance>6626.0703125</wpml:distance>
 <wpml:duration>1802.47204589844</wpml:duration>
 <wpml:autoFlightSpeed>8</wpml:autoFlightSpeed>
 <Placemark>
   <Point>
      <coordinates>
       117.708960407258,39.0567143754473
     </coordinates>
    <wpml:index>0</wpml:index>
    <wpml:executeHeight>100</wpml:executeHeight>
    <wpml:waypointSpeed>8</wpml:waypointSpeed>
    <wpml:waypointHeadingParam>
      <wpml:waypointHeadingMode>manually</wpml:waypointHeadingMode>
      <wpml:waypointHeadingAngle>0</wpml:waypointHeadingAngle>
      <wpml:waypointPoiPoint>0.000000,0.000000,0.000000</wpml:waypointPoiPoint>
      <wpml:waypointHeadingAngleEnable>0</wpml:waypointHeadingAngleEnable>
    </wpml:waypointHeadingParam>
    <wpml:waypointTurnParam>
      <wpml:waypointTurnMode>toPointAndStopWithDiscontinuityCurvature</wpml:waypointTurnMode>
      <wpml:waypointTurnDampingDist>0</wpml:waypointTurnDampingDist>
    </wpml:waypointTurnParam>
    <wpml:useStraightLine>1</wpml:useStraightLine>
      <wpml:actionGroupId>0</wpml:actionGroupId>
      <wpml:actionGroupStartIndex>0</wpml:actionGroupStartIndex>
      <wpml:actionGroupEndIndex>0</wpml:actionGroupEndIndex>
      <wpml:actionGroupMode>sequence</wpml:actionGroupMode>
      <wpml:actionTrigger>
        <wpml:actionTriggerType>reachPoint</wpml:actionTriggerType>
      </wpml:actionTrigger>
      <wpml:action>
        <wpml:actionId>0</wpml:actionId>
       <wpml:actionActuatorFunc>takePhoto</wpml:actionActuatorFunc>
        <wpml:actionActuatorFuncParam>
          <wpml:payloadPositionIndex>0</wpml:payloadPositionIndex>
```

### 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

```
val guid = System.currentTimeMillis()/ 1000

missionManager =
DeviceManager.getDeviceManager().getFirstDroneDevice()?.getWayPointMissionManager()

missionManager?.uploadKmzMissionFile(kmzFilePath, guid.toInt(), object:
CommonCallbacks.CompletionCallbackWithProgressAndParam<Long> {
    override fun onProgressUpdate(progress: Double) {}
    override fun onSuccess(guid: Long?) { }
    override fun onFailure(error: IAutelCode, msg: String?) {}
}
```

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