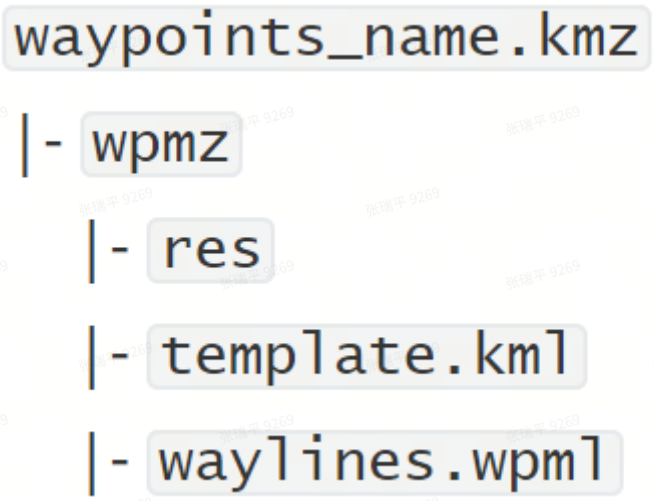
KMZ Task File Explanation

Ⅰ、KMZ File Explanation

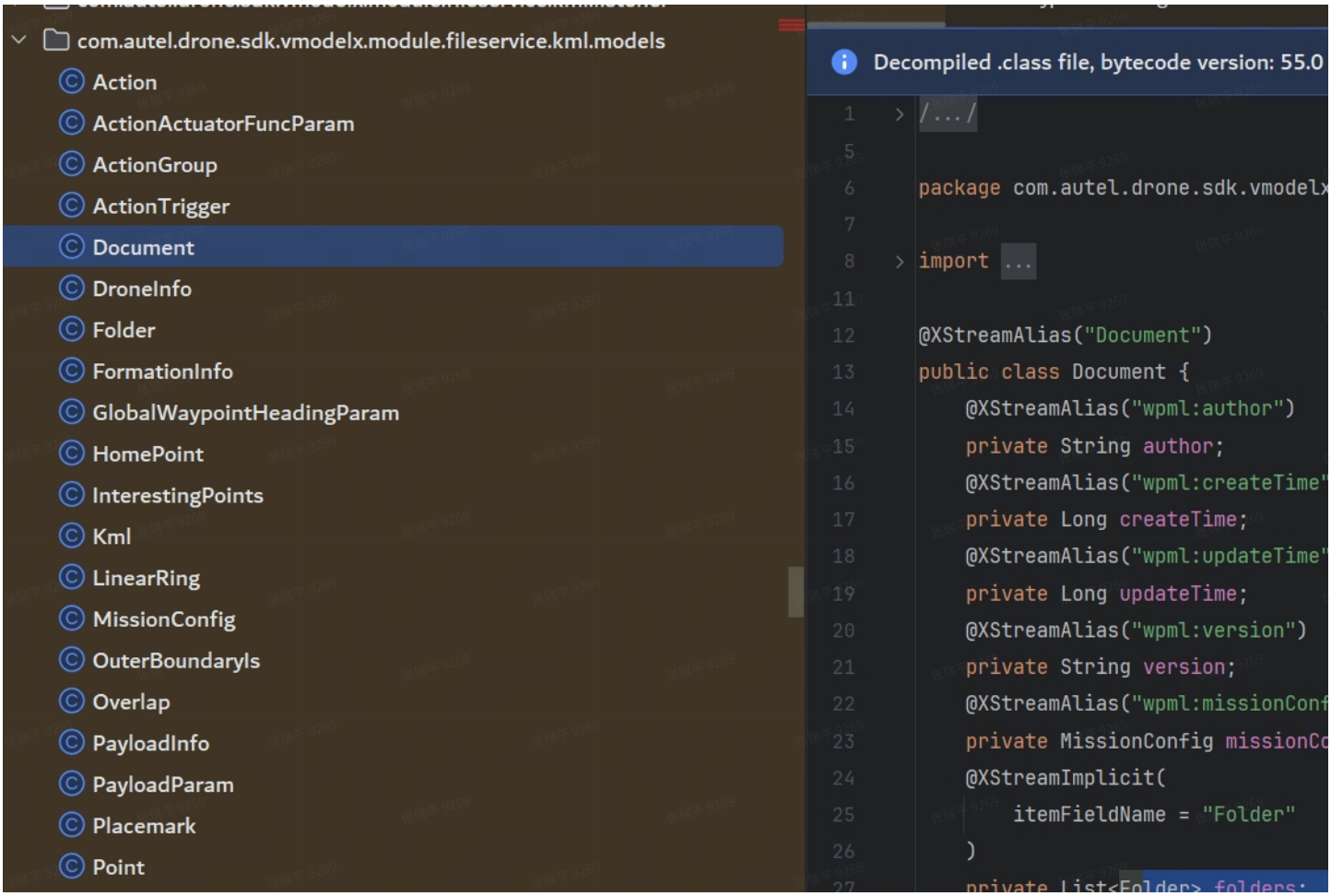


WPML (WayPoint Markup Language) File Format

WPML (WayPoint Markup Language) is a file format standard for waypoint files, which is an extension of the KML (Keyhole Markup Language) definition. WPML waypoint files follow the KMZ archiving requirements and all waypoint files end with the ".kmz" suffix. The WPML waypoint file format serves as a carrier for digital waypoint assets.

1. Essentially a ZIP Compressed File: Both KML and WPML are XML document formats.
2. Template File: The `template.kml` file is referred to as a "template file," which facilitates user editing and planning.
3. Execution File: The `waylines.wpml` file is known as an "execution file."
4. RES Resource Folder: The RES folder contains auxiliary resources required for the waypoints, such as reference target photos prepared in advance before precise rephotography.

Ⅱ、MSDK Support for KMZ



1.Definition of the KML Object Model

2.MissionConfig as Global Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:wpml="http://www.dji.com/wpmz/1.0.0">
  <Document>
    <wpml:missionConfig>
    <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>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
      <Placemark>
```

3.Placemark Represents a Waypoint

Point Represents Geographic Coordinate

ActionGroup as a Collection of Actions

Action Represents a Specific Action，such as take photo

```
<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>
    </Point>
    <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:actionGroup>
      <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>
```

III、KML Objects to XML Document

1.Converting KML Objects to XML Text Using XStream

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

2.To create `template.kml` and `waylines.wpml` files, populate them with data, and then package them into a KMZ file

IV、KMZ Task Execution And Control

1.Uploading KMZ Files to the Aircraft and Executing KMZ Tasks

```
    Uploading KMZ

    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?) {}
})
```

```
Task control: (param is MissionKmlGUIDBean)
 missionManager?.startMission(MissionKmlGUIDBean(it.toInt()),
    object : CommonCallbacks.CompletionCallbackWithParam<Void> {
        override fun onFailure(error: IAutelCode, msg: String?) {}
        override fun onSuccess(t: Void?) {}
  })


 val isKml = true
 missionManager?.pauseMission(object: CommonCallbacks.CompletionCallbackWithParam<Void>{
    override fun onFailure(error: IAutelCode, msg: String?) {}
    override fun onSuccess(t: Void?) {}
}, isKml)

...
```

V、Reference

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

2.Demo MSDK2.0-MissionDemo