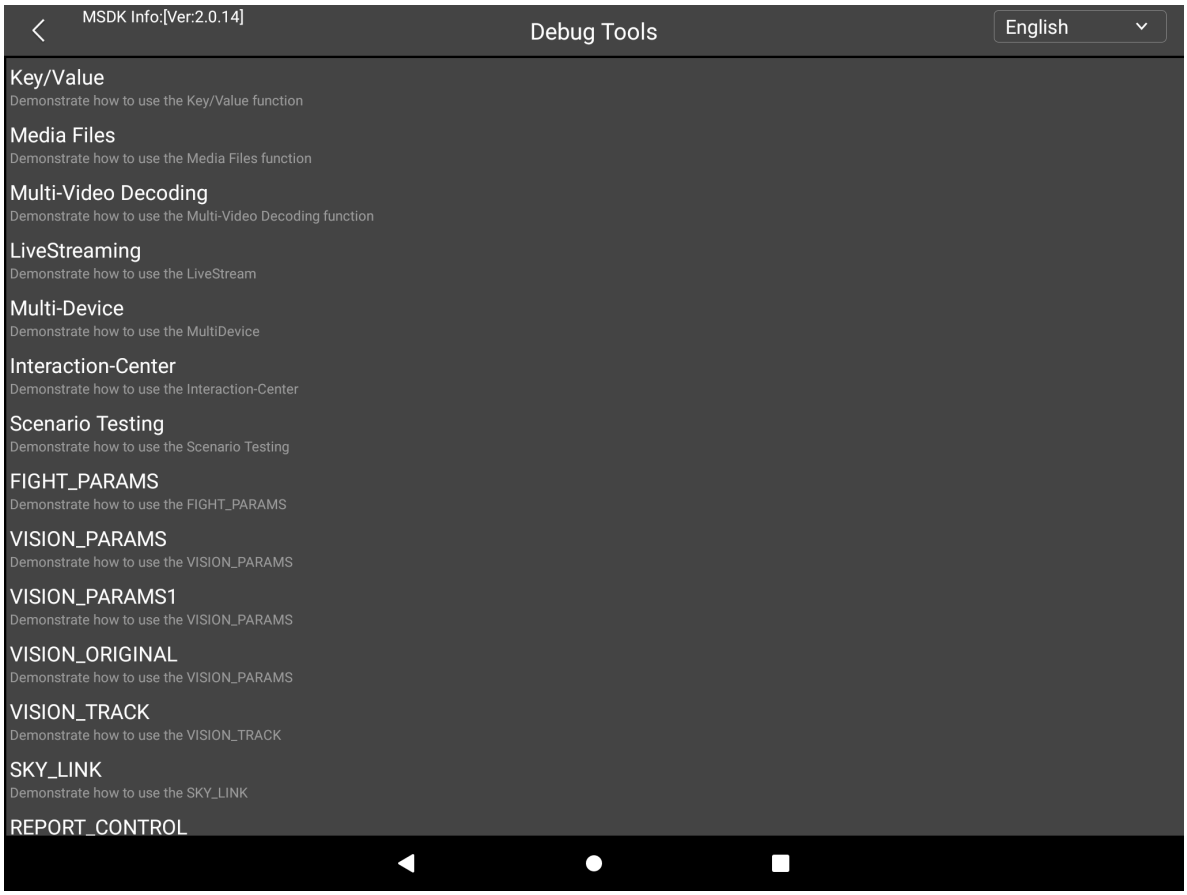
Debug Tools provides API debugging tools for easy parameter access and behavior control of aircraft hardware modules.



- Key/Value: debugs all keys supported by aircraft.
- Media Files: uploads, downloads, queries, and deletes mission files; obtains, deletes, and downloads video files; performs health checks.
- Multi-Video Decoding: decodes videos in different formats at the same time.
- LiveStreaming: plays back real-time video streams.
- Scenario Testing: controls remote controller connection, remote controller/IMU/compass/gimbal calibration, and photo and video recording.

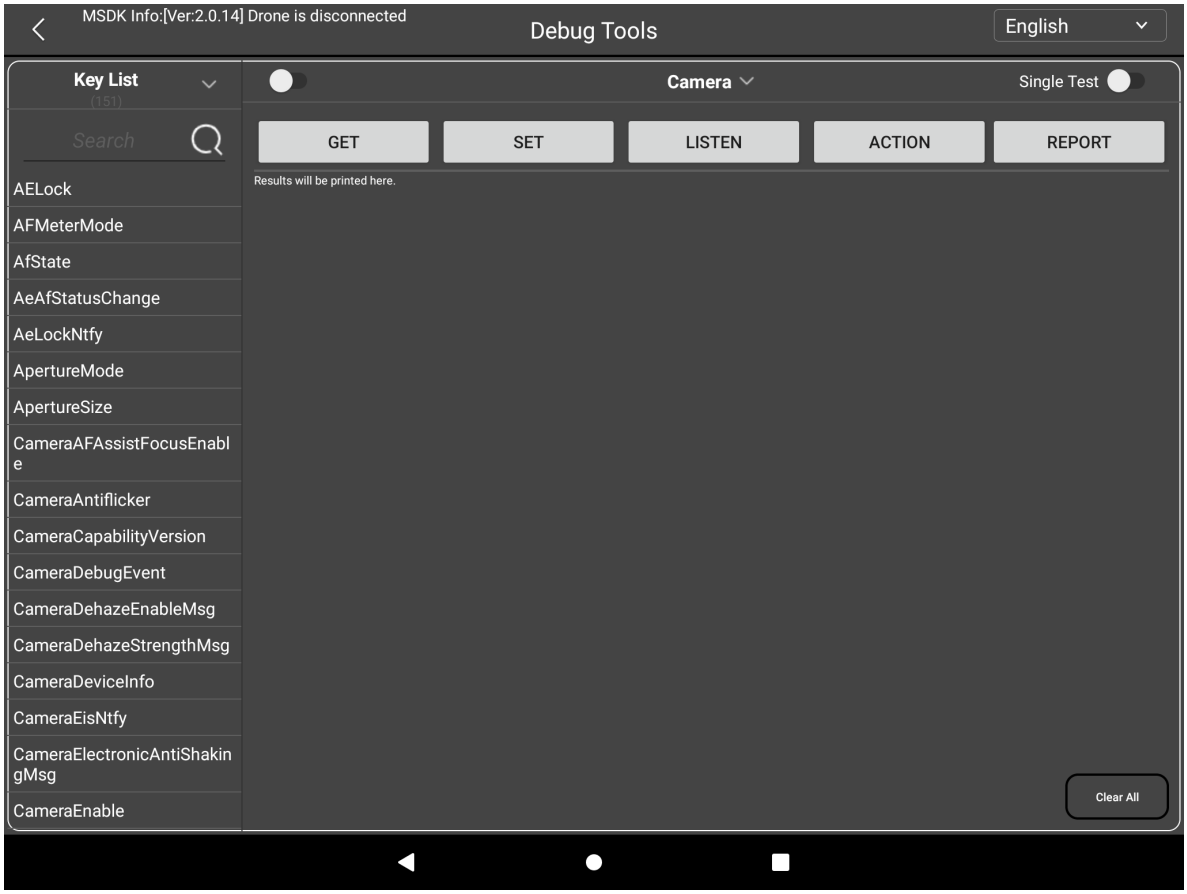
Key/Value

Overview

The KeyManager class provides a set of methods to quickly access the parameters and control the behavior of hardware modules. First, the **createKey** method provided by KeyTools is used to create Autel keys. Then the **Set**, **Get**, **Action**, **Listen** and **Report** methods initiated by KeyManager can be used to control the hardware modules.

Debugging

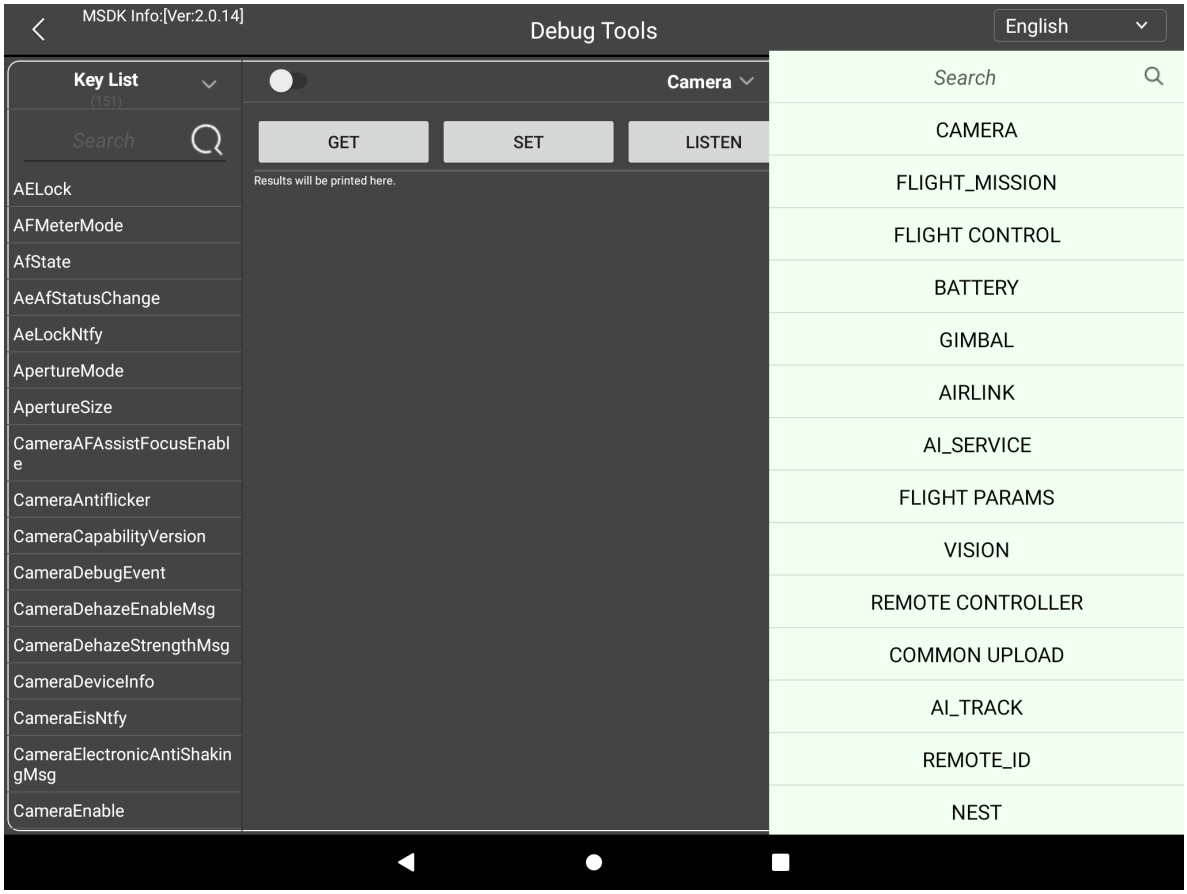
The following figure displays the Key/Value debugging page, where the camera module is set by default.



1. The right pane displays the **Camera** module selected. You can click and switch to another module.
2. **Key List** on the left is a list of all keys supported by the currently selected module. You can search directly or by page.

Modules supporting debugging: Camera, FLGHT_MISSION, FLIGHT CONTROL, BATTERY, GIMBAL, AIRLINK, CHANNEL_TYPE_NETMESH, AI_SERVICE, FLIGHT PARAMS, VISION, REMOTE CONTROLLER, COMMON UPLOAD, AI_TRACK, REMOTE_ID, NEST, SystemManager, MISSION MANAGER, ACCESSORIES PROXY, Autonomy

The following figure shows how to switch to another module.

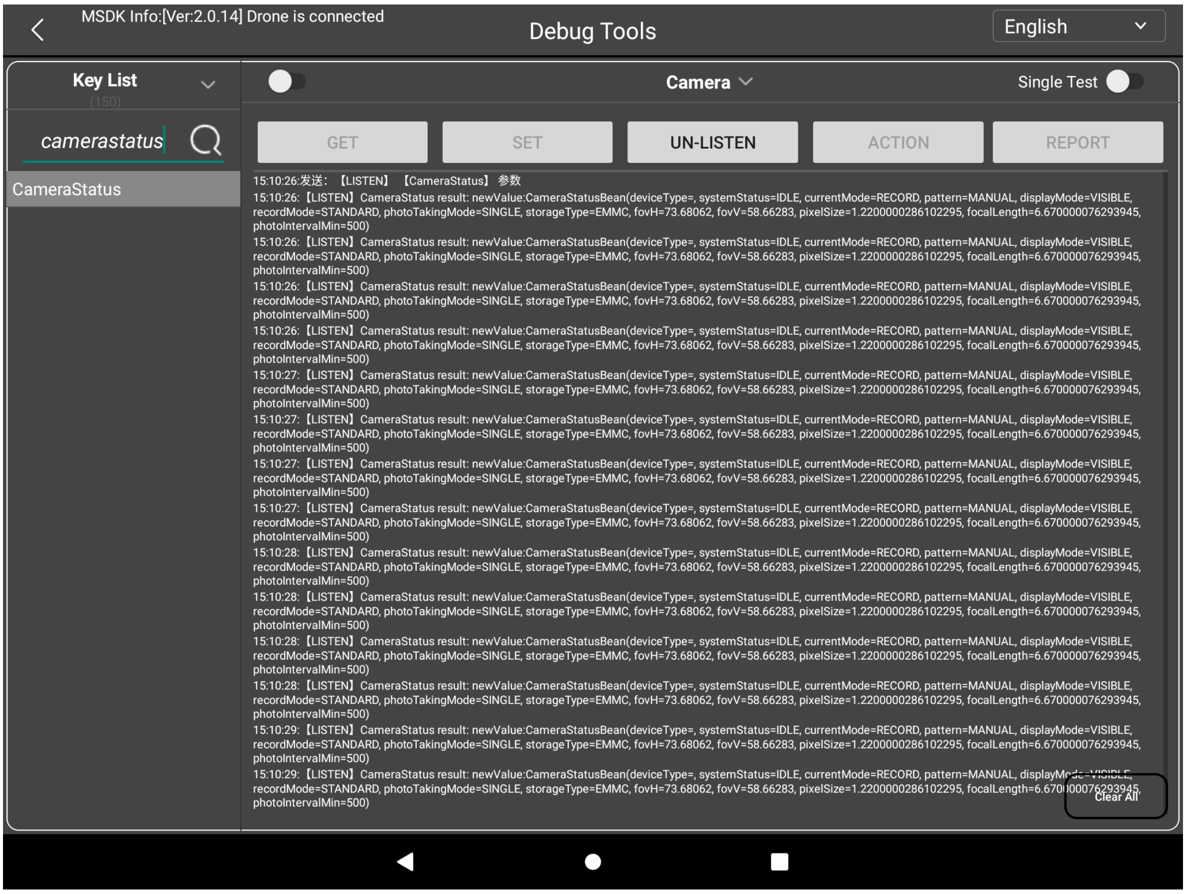


Listen

(Example) Querying Camera Status

Select **CameraStatus** in the key list. Because **CameraStatus** supports only monitoring, the **Get**, **Set**, **Action**, and **Report** options are grayed out. Click **Listen**, and the parameters of **CameraStatus** will be displayed. At the same time, **Listen** will change to **UnListen** and the parameters will not be displayed. If you click **UnListen**, listening will be canceled and the parameters will be hidden.

The following figure shows the result of successful calling of the **Listen** method.



Sample

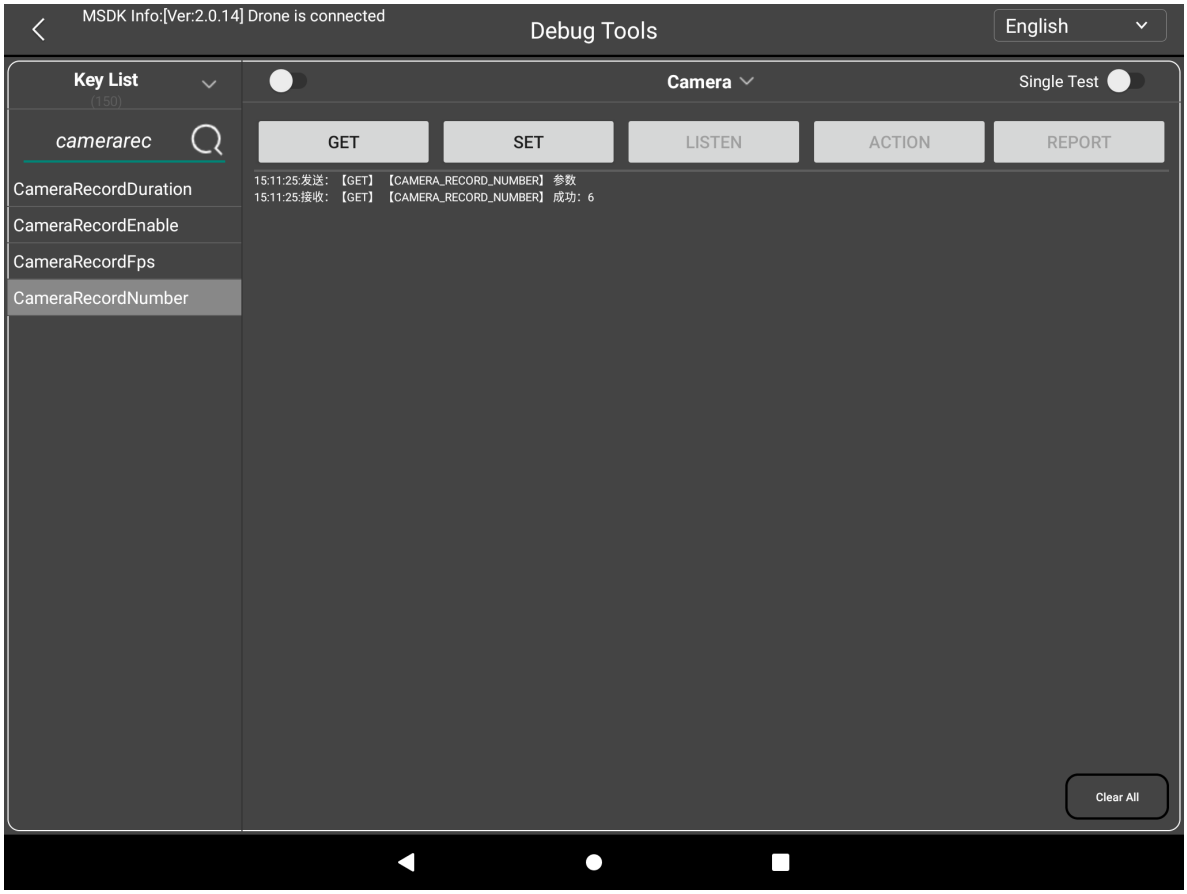
```
fun addCameraStatusListen() {
    val keyCameraStatus: AutelKey<CameraStatusBean> =
KeyTools.createKey(CameraKey.KeyCameraStatus)

 DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.listen
(keyCameraStatus, cameraStatusCallback)
}

private val cameraStatusCallback = object :
CommonCallbacks.KeyListener<CameraStatusBean> {
    override fun onValueChange(oldValue: CameraStatusBean?, newValue:
CameraStatusBean) {
    }
}
```

Get

(Example) Querying The Number Of Recordings

Select **CameraRecordNumber** in the key list. Because **CameraRecordNumber** supports only **Get** and **Set**, the **Listen**, **Action**, and **Report** methods are grayed out. Click **Get**, and the parameters of **CameraRecordNumber** will be displayed.

The following figure shows the result of successful calling of the **Get** method.



Sample

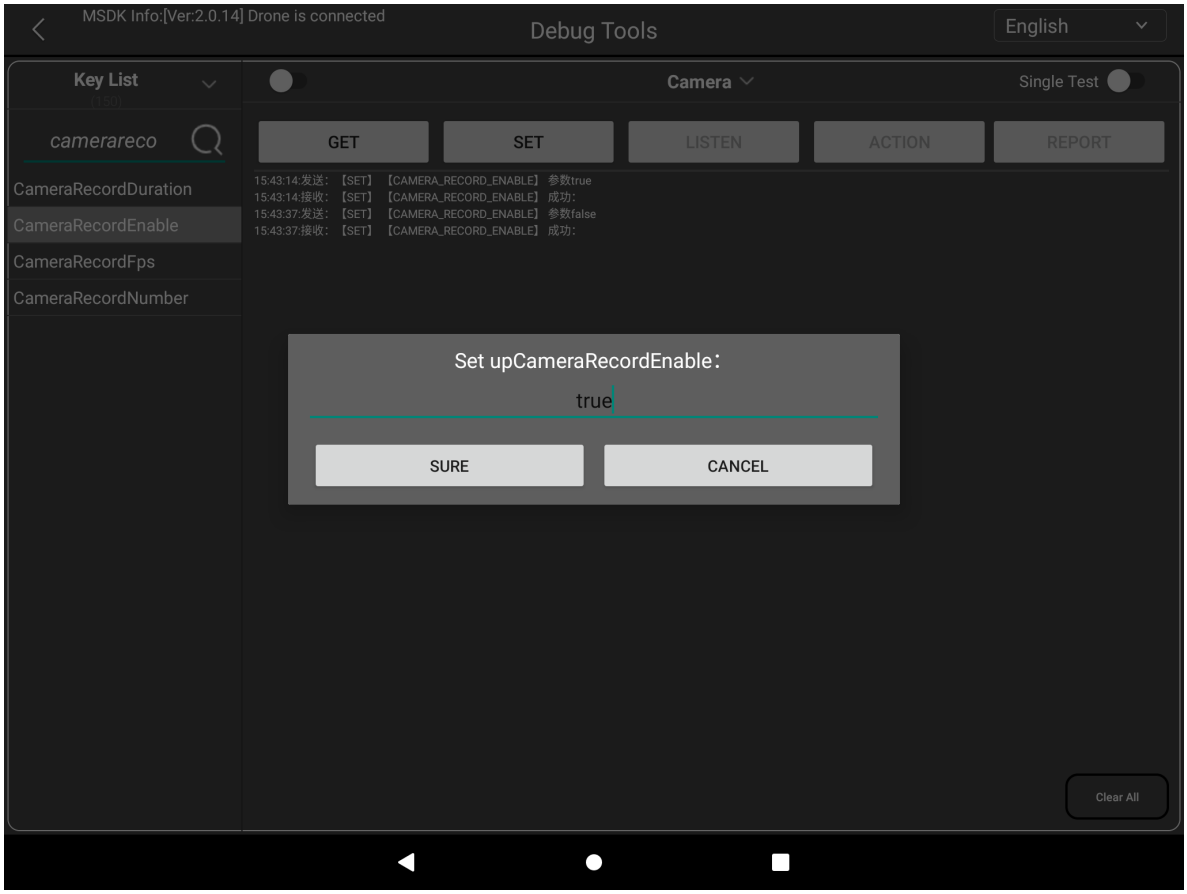
```
fun getCameraRecordNumber() {
        val keyCameraRecordNumber: AutelKey<Int> =
KeyTools.createKey(CameraKey.KeyCameraRecordNumber)

 DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.getVal
ue(keyCameraRecordNumber,
        object : CommonCallbacks.CompletionCallbackWithParam<Int> {
            override fun onSuccess(t: Int?) {

            }

            override fun onFailure(error: IAutelCode, msg: String?) {

            }
        })
    }
```

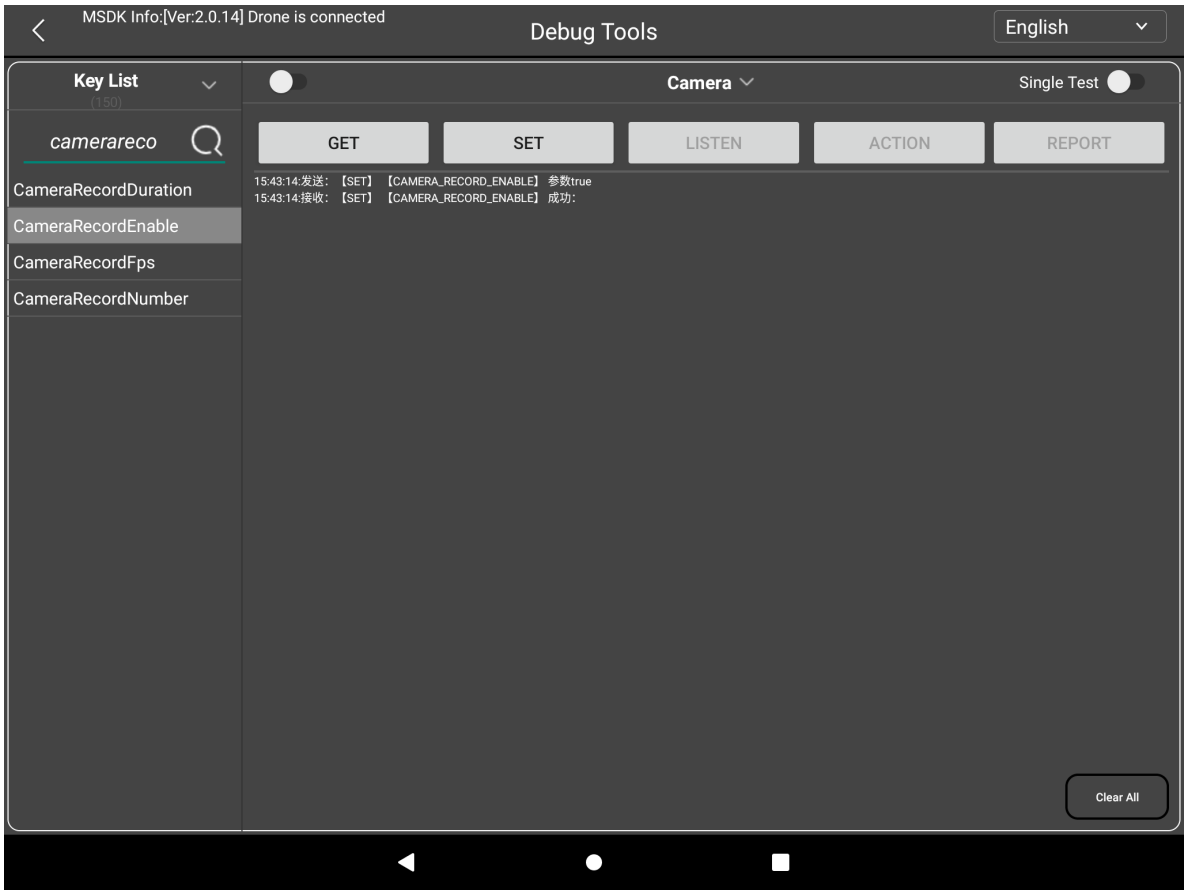
Set

(Example) Enabling Recording

Select **CameraRecordEnable** in the key list. Because **CameraRecordEnable** supports only **Get** and **Set**, the **Listen**, **Action**, and **Report** methods are grayed out. Click **Set**, and the request result (success or failure) will be displayed.

The following figure shows the configuration and successful calling of the **Set** method.





Sample

```
fun setRecordEnable(open: Boolean) {

 DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.setVal
ue(
        KeyTools.createKey(CameraKey.KeyCameraRecordEnable), open,
```

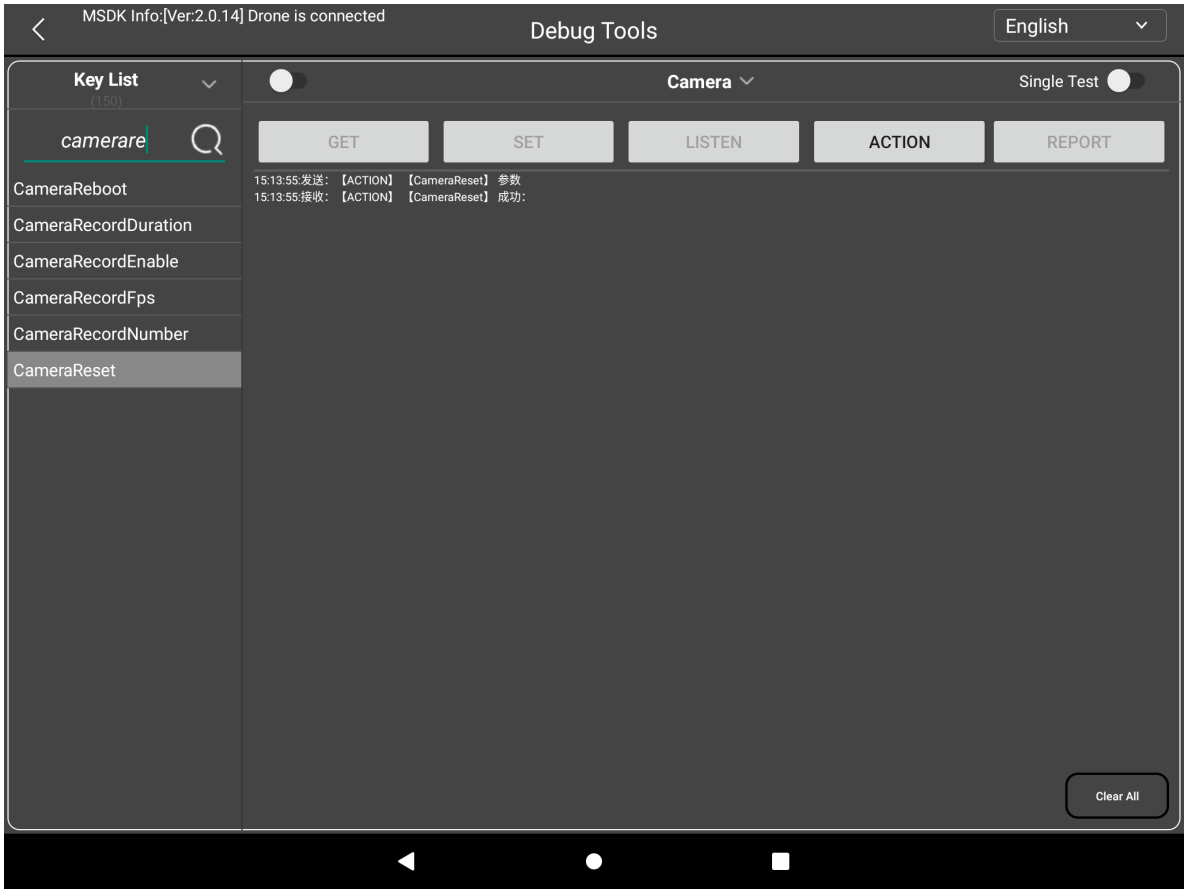
```
        object : CommonCallbacks.CompletionCallback {
            override fun onSuccess() {

            }

            override fun onFailure(code: IAutelCode, msg: String?) {

            }

        }
    )
}
```

Action

(Example) Resetting Camera

Select **CameraReset** in the key list. Because **CameraReset** supports only **Action**, the **Get**, **Set**, **Listen**, and **Report** methods are grayed out. Click **Action**, and the request result (success or failure) will be displayed.

The following figure shows the result of successful calling of the **Action** method.



Sample

```
fun cameraReset() {
    val key = KeyTools.createKey(CameraKey.KeyCameraReset)

 DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.perfor
mAction(key, null,
    object : CommonCallbacks.CompletionCallbackWithParam<Void> {
        override fun onSuccess(t: Void?) {

        }

        override fun onFailure(code: IAutelCode, msg: String?) {
        }
    })
}
```