

WizarViewAgent External Interface Description

Version : 0.0.9

目录

1. Overview	3
1.1 Document Description	3
2. Interface Definition and Explanation	3
2.1 Refreshing Wizarview Remote Configuration Information	3
2.2 Querying Configured Application Information in Wizarview	4
2.3 Downloading Specified Configured Applications	6
2.3.1 Downloading Applications with Specified AppID	6
2.3.2 Download Completion Broadcast	8
2.4 Querying the Current Download Progress by AppID	10
2.5 Obtaining the Configuration Status of an Application by	11
2.6 Obtaining the Configuration Status of an Application by PackageName	12
2.7 Obtaining Terminal Registration Address via AIDL Interface	13
2.8 Querying Terminal Status via Properties	14
2.9 Registering Terminal with Wizarview	16
2.10 Submitting Logs from Terminal	18
2.11 Third-Party Application Message Push Reception	20
3 Interface Calling Process	22
3.1 Aidl Interface	22
3.1.1 Aidl Interface Definition	22
3.1.2 Obtaining an AIDL Instance Object	23
3.2 Constants Used in the Interface	24

1. Overview

1.1 Document Description

This document is primarily intended for third-party applications to obtain the latest configuration information on Wizarview. The document requirements and included content are:

1. Clearly define the semantics of the interface and describe how to call it.
2. Clearly define the required applications.

Note: Applications needed during development: wizarviewAgent version 2.0.0 or above.

2. Interface Definition and Explanation

2.1 Refreshing Wizarview Remote Configuration Information

Interface semantic definition description:

Used for the WizarviewAgent application to actively obtain the latest configuration application list from Wizarview, belonging to a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

```
int refreshAppList();
```

Parameter explanation:

None

Return value:

int

Return value	Meaning
0	Update completed
-1	Network not connected error
-2	Terminal not registered error
-3	Terminal unable to log in to the server error
-4	Terminal unable to connect to the server error
-5	Downloader request to refresh list timeout (10s)!
-6	Currently downloading

Calling method:

Called using AIDL. For details, see 3.1

2.2 Querying Configured Application Information in Wizarview

Interface semantic definition description :

Used to obtain the configured application list through the interface of the WizarviewAgent application, belonging to a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

```
AppInfo [] queryAppInfos(int installType, int status, int appType );
```

Parameter explanation:

Type	Value	Meaning
installType	INSTALL_TYPE_OPTION	Optional configuration
	INSTALL_TYPE_REQUIRED	Mandatory configuration
	INSTALL_TYPE_REQUIRED_SILENT	Silent installation download configuration
	INSTALL_TYPE_ALL	All configuration items
status	STATUS_STOP	Application not downloaded
	STATUS_DOWNLOADING	Application is downloading
	STATUS_DOWNLOADED	Application downloaded
	STATUS_INSTALLED	Application installed
	STATUS_TYPE_ALL	All status items
appType	APPTYPE_APK	Configuration item type is only apk
	APPTYPE_FIRMWARE	Configuration item type is only firmware
	APPTYPE_PARAMETER	Configuration item type is

		only parameter file
	APPTYPE_TYPE_ALL	All type items

Note:

For constant definitions, see 3.2

Return value:

AppInfo[] appInfos or null.

Calling method:

Called using AIDL. For details, see 3.1

2.3 Downloading Specified Configured Applications

This interface is divided into two parts, one part is how to download, and the other part is how to obtain the download result or installation result.

2.3.1 Downloading Applications with Specified AppID

Interface semantic definition description:

Used to notify the WizarviewAgent application to download specified applications, the interface is called using AIDL, belonging to an asynchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

Int downloadAppInfoByAppID(int appID);

Parameter explanation:

Int appID: The AppIDs to be downloaded.

Return value:

Return value	Meaning
0	Start downloading
-1	Network not connected error
-2	Terminal not registered error
-3	Terminal unable to log in to the server error
-4	Terminal unable to connect to the server error
-5	Parameter error
-6	Currently downloading
-7	Download completed
-8	Installed

Note:

Server errors are not listed temporarily

Calling method:

See 3.1

2.3.2 Download Completion Broadcast

Interface semantic definition description:

Transmits the download completion or download error information to third-party applications through broadcast, belonging to an asynchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS


Interface description:

Use BroadcastReceiver to listen to download progress, and obtain the current appid's download progress through data passed in the intent.

Calling method:

1: Register to listen and declare permission

```
<!-- Declare permission to listen to Download broadcast -->
```

```
 android:permission="android.permission.CLOUDPOS_DOWNLOADRECEI  
VER"
```

Code example:

```
<receiver                android:name="com.xxx.xxx.receiver.XXReceiver"  
  
    android:permission="android.permission.CLOUDPOS_DOWNLOADRECEI  
    VER">  
  
<!-- Declare the filter condition for listening to Download broadcast -->  
  
<action  
  
    android:name="android.intent.action.RESULT_OF_DOWNLOAD"/>
```


</receiver>

2: Get callback information

```
public class XXReceiver extends BroadcastReceiver {  
  
    @Override  
  
    public void onReceive(Context context, Intent intent) {  
  
        // Obtain the current download information callback through Intent.  
  
        String value = intent.getStringExtra(key)  
  
    }  
  
}
```

Parameter explanation:

None

Return value:

Name	Value	Meaning
percent	100	Download complete
	0	Not downloaded
appid	1245	Not downloaded
status	0x00000001 (STATUS_STOP)	Download failed
	0x00000004 (STATUS_DOWNLOADED)	Download complete

2.4 Querying the Current Download Progress by AppID

Interface semantic definition description:

Obtains the current download progress of a specified AppID from the WizarviewAgent application, belonging to a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition: Unloaded status

```
String queryAppInfoDownloadProgress(int appID);
```

Parameter explanation:

Int appID: The AppID of the download file.

Return value:

String: In jsonobject format. appID=?,percent=?,status=?,speed=? or null
(Invalid AppID)

Name	Value	Meaning
percent	0 ~ 100	Download progress
status	STATUS_STOP STATUS_DOWNLOADING STATUS_DOWNLOADED STATUS_INSTALLED	Current application status
appID	123456	App ID
speed	123456789	Average download speed

Note:

For specific definitions of status constants, see 3.2

Calling method:

Called using AIDL. *Note: For specific calling, see 3.1

2.5 Obtaining the Configuration Status of an Application by

AppIDInterface semantic definition description:

Obtains the configuration information under a specified AppID from the interface of the WizarviewAgent application, belonging to a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

```
AppInfo queryAppInfoByAppID(int appID);
```

Parameter explanation:

Int appID: The AppID of the configuration item.

Return value:

appInfo or null (not found or parameter error).

Calling method:

Called using AIDL.

Note:

For specific calling, see 3.1

2.6 Obtaining the Configuration Status of an Application by PackageName

Interface semantic definition description:

Obtains the configuration status of an application from the WizarviewAgent application by package name, belonging to a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

```
AppInfo[] queryAppInfoByPackageName(String packageName, int appType);
```

Parameter explanation:

String packageName: The package name of the Android application configuration.

Int appType:

appType	APPTYPE_APK	Configuration item type is only apk
	APPTYPE_FIRMWARE	Configuration item type is only firmware
	APPTYPE_PARAMETER	Configuration item type is only parameter file

	APPTYPE_TYPE_ALL	All type items
--	------------------	----------------

Return value:

AppInfo[] appInfos or null.

Calling method:

Called using AIDL.

Note:

For specific calling, see 3.

2.7 Obtaining Terminal Registration Address via AIDL Interface

Interface semantic definition description:

Obtains the application's registration address from the WizarviewAgent application, a synchronous method.

Required permission:

CLOUDPOS_ACCESS_TMS

Interface definition:

String getRegisterAddress();

Parameter explanation:

None

Return value:

String registration address.

Calling method:

Called using the AIDL method.

Note:

For specific calling, see 3.1

2.8 Querying Terminal Status via Properties

Calling method:

Query the value of "wp.wizarview.status" by querying system properties.

Interface function:

Queries the current terminal status: whether the terminal is registered or whether the terminal is online.

Interface calling process or characteristics:

A synchronous query operation.

Defined interface:

1: Parameters Key:

wp.wizarview.status

2: Return results:

The result is an int.

Return result	Meaning
0	Online
-1	No network on the current terminal
-2	Terminal not registered

-3	Terminal unable to login to the server
-4	Terminal service unable to connect to the server

Note:

If the terminal does not support this function, the terminal will not have the "wp.wizarview.status" property. 3: Explanation of the calling method (1): Use SystemProperties.get("wp.wizarview.status", "unknown") to obtain the property.

(2): Since SystemProperties.get() is a hidden API, it can be called through reflection, see the calling method example for the method.

Interface calling example

// Return null if the property does not exist

```
public static String getTerminalStatus() {
```

```
    String key = "wp.wizarview.status";
```

```
    String defaultValue = "unknown";
```

```
    Object property = null;
```

```
    try {
```

```
        Class<?> systemProperties = Class
```

```
            .forName("android.os.SystemProperties");
```

```
        Log.i("systemProperties", systemProperties.toString());
```

```
        property = systemProperties
```

```
            .getMethod("get", new Class[]{String.class, String.class})
```

```
            .invoke(systemProperties, new Object[]{key, defaultValue});
```

```

    } catch (Exception e) {

        e.printStackTrace();

    }

    if (defaultValue.equals(property.toString())) {

        Log.e("getTerminalStatus", "not found wp.wizarview.status.");

        return null;

    }

    return property.toString();

}

```

2.9 Registering Terminal with Wizarview

Calling method:

Start the wizaragent via broadcast and have the terminal register with Wizarview.

Interface function:

Starts the wizaragent and registers the terminal with Wizarview.

Interface calling process or characteristics:

Does not return a calling result; registration will not continue if the terminal has already been registered.

Defined interface:

1. Parameters

Key	Command	Meaning
wizar_cmd	Register_wizarview	Start Wizarviewagent, and register terminal to Wizarview(No Return)

2. Return Result

No return results provided.

3. Explanation of the calling method

The calling method uses the default Android broadcast method.

(1) Specify the package name in Android's Intent.

`com.wizarpos.wizarviewagent"`

(2) Add action in Android's Intent

`"android.intent.action.WIZARVIEWAGENT"`

(3) Add the command to be passed in Android's Intent

`"wizar_cmd": "register_wizarview"`

Interface calling example

```
private static final String CMD_REGISTER_WIZARVIEW = "register_wizarview";
```

```
private static final String ACTION_WIZARVIEWAGENT =
```

```
    "android.intent.action.WIZARVIEWAGENT";
```

```
private static final String CMD_WIZARVIEWAGENT = "wizar_cmd";
```

```

// Send broadcast

Intent intent = new Intent();

intent.setAction(ACTION_WIZARVIEWAGENT);//Set the filter condition.

// Pass command parameters

intent.putExtra(CMD_WIZARVIEWAGENT, CMD_REGISTER_WIZARVIEW);

intent.setPackage("com.wizarpos.wizarviewagent");// Specify the
package name.

this.sendBroadcast(intent);

```

2.10 Submitting Logs from Terminal

Calling method:

Start the wizaragent via broadcast and have the terminal submit logs to Wizarview.

Interface function:

Starts the wizaragent and submits logs from the terminal to Wizarview.

Interface calling process or characteristics:

Does not return a calling result; submitting logs may fail.

Defined interface:

1: Parameters

Name	Value	Meaning
wizar_cmd	submit_log	Submit logs to Wizarview

--	--	--

2: Return results

No return results provided

3: Explanation of the calling method

The calling method uses the default Android broadcast method.

(1) Specify the package name in Android's Intent.

"com.wizarpos.wizarviewagent"

(2) Add action in Android's Intent

"android.intent.action.WIZARVIEWAGENT"

(3) Add the command to be passed in Android's Intent

"wizar_cmd": "submit_log"

Interface calling example

```
private static final String CMD_SUBMIT_LOG = "submit_log";
```

```
private static final String ACTION_WIZARVIEWAGENT =
```

```
"android.intent.action.WIZARVIEWAGENT";
```

```
private static final String CMD_WIZARVIEWAGENT = "wizar_cmd";
```

```
//Send broadcast
```

```
Intent intent = new Intent();
```

```
intent.setAction(ACTION_WIZARVIEWAGENT); //Set the filter condition.
```

```
//Pass command
```

```
intent.putExtra(CMD_WIZARVIEWAGENT, CMD_SUBMIT_LOG);
```

```
//Specify the package name.
```

```
intent.setPackage("com.wizarpos.wizarviewagent");
```

```
this.sendBroadcast(intent);
```

2.11 Third-Party Application Message Push Reception

Interface semantic definition description:

Third-party applications receive push messages from remote servers.

Permission:

android.permission.CLOUDPOS_PUSHSERVICE

Calling method:

Third-party applications receive server push messages by setting up a message broadcast receiver.

Interface functionality:

Delivers information to third-party applications via the server.

Interface calling process or characteristics:

This interface involves passive reception callbacks and does not require a return of call results.

Defined interface:

1: Message format

The message is stored in the intent of the broadcast. Values of relevant information are obtained through the KEY.

KEY	Meaning
-----	---------

notification	Content pushed by the server
time	Time of the server push. (Requires wizarviewagent version 3.0 or above)
appid	appid ID of the server information. (Requires wizarviewagent version 3.0 or above)

2: Return results

No return results provided.

3: Explanation of the calling method

- (1) If the application has not correctly set up the message broadcast, it will not receive push information.
- (2) If the application has not declared permission, it will not receive push information.
- (3) If the application has never been activated, it will not receive push information.

Interface calling example

AndroidManifest :

```
<uses-permission android:name="android.permission.CLOUDPOS_PUSHSERVICE"/>
```

```
<receiver
```

```
    android:name="com.wizarpos.test.receiver.MessageReceiver"
```

```
    android:label="@string/app_name" >
```

```
    <intent-filter>
```

```
        <action android:name="com.wizarpos.test" />
```

```
    </intent-filter>
```

```
</receiver>
```

Broadcast Receiver :

```
public class MessageReceiver extends BroadcastReceiver {  
  
private static final String KEY = "notification";  
  
@Override  
  
public void onReceive(Context context, Intent intent) {  
  
String msg = intent.getStringExtra(KEY);  
  
MainActivity.writerInSuccessLog(msg);;  
  
}  
  
}
```

3 Interface Calling Process

3.1 Aidl Interface

3.1.1 Aidl Interface Definition

Directory and Filename of the Interface

com.wizarpos.wizarviewagent.service.aidl

src/com/wizarpos/wizarviewagent/aidl/AppInfo.java

src/com/wizarpos/wizarviewagent/aidl/AppInfo.aidl

src/com/wizarpos/wizarviewagent/aidl/IWizarviewService.aidl

```

package com.wizarpos.wizarviewagent.aidl;

import com.wizarpos.wizarviewagent.aidl.AppInfo;

interface IWizarviewService{

    int refreshAppList();

    int downloadAppInfoByAppID(int appID);

    AppInfo [] queryAppInfos(int installType, int status, int appType );

    AppInfo queryAppInfoByAppID(int appID);

    AppInfo[] queryAppInfoByPackageName(String packageName, int appType);

    String queryAppInfoDownloadProgress(int appID);

}

```

3.1.2 Obtaining an AIDL Instance Object

```

//bind service

public boolean bindAidlService(Context context){

    Intent intent = new Intent();

    ComponentName comp = new ComponentName(

        "com.wizarpos.wizarviewagent",

        "com.wizarpos.wizarviewagent.service.WizarviewService");

    intent.setComponent(comp);

    ServiceConnectionImpl connection = new ServiceConnectionImpl();

    boolean isSuccess = contex.bindService(intent, connection,

Context.BIND_AUTO_CREATE);

```

```

        return isSuccess;
    }

    // AIDL connection callback

    private class ServiceConnectionImpl implements ServiceConnection {

        @Override

        public void onServiceConnected(ComponentName name, IBinder service) {

            // Obtain the interface object

            IWizarviewService            aidlService            =

            IWizarviewService.Stub.asInterface(service);

        }

        @Override

        public void onServiceDisconnected(ComponentName name) {

        }

    }
}

```

3.2 Constants Used in the Interface

Configured Application Installation Types

Data Type	Type Name	Value	Meaning
int	INSTALL_TYPE_OPTIONAL	0x00000001	Optional download type
int	INSTALL_TYPE_REQUIRED	0x00000002	Mandatory download type
int	INSTALL_TYPE_REQUIRED_SILENT	0x00000004	Silent installation download type
int	INSTALL_TYPE_ALL	0xFFFFFFFF	All installation types

Configured Application States

Data Type	Type Name	Value	Meaning
int	STATUS_STOP	0x00000001	Not downloaded state
int	STATUS_DOWNLOADING	0x00000002	Currently downloading state
int	STATUS_DOWNLOADED	0x00000004	Downloaded state
int	STATUS_INSTALLED	0x00000008	Installed state
int	STATUS_TYPE_ALL	0xFFFFFFFF	All application states

Configured Application Types

Data Type	Type Name	Value	Meaning
int	APPTYPE_APK	0x00000001	APK application type
int	APPTYPE_FIRMWARE	0x00000002	Firmware application type
int	APPTYPE_PARAMETER	0x00000004	Parameter application type
int	APPTYPE_TYPE_ALL	0xFFFFFFFF	All application types