

---

# The OpenCV Manager Manual

*Release 2.4.3*

October 24, 2012



# CONTENTS

<b>1</b>	<b>Android OpenCV Manager</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Manager Workflow . . . . .	3
1.3	Java OpenCV Loader . . . . .	5
1.4	Base Loader Callback Interface implementation . . . . .	6
1.5	Loader Callback Interface . . . . .	8
1.6	Install Callback Interface . . . . .	9
<b>2</b>	<b>Java API</b>	<b>11</b>



# ANDROID OPENCV MANAGER

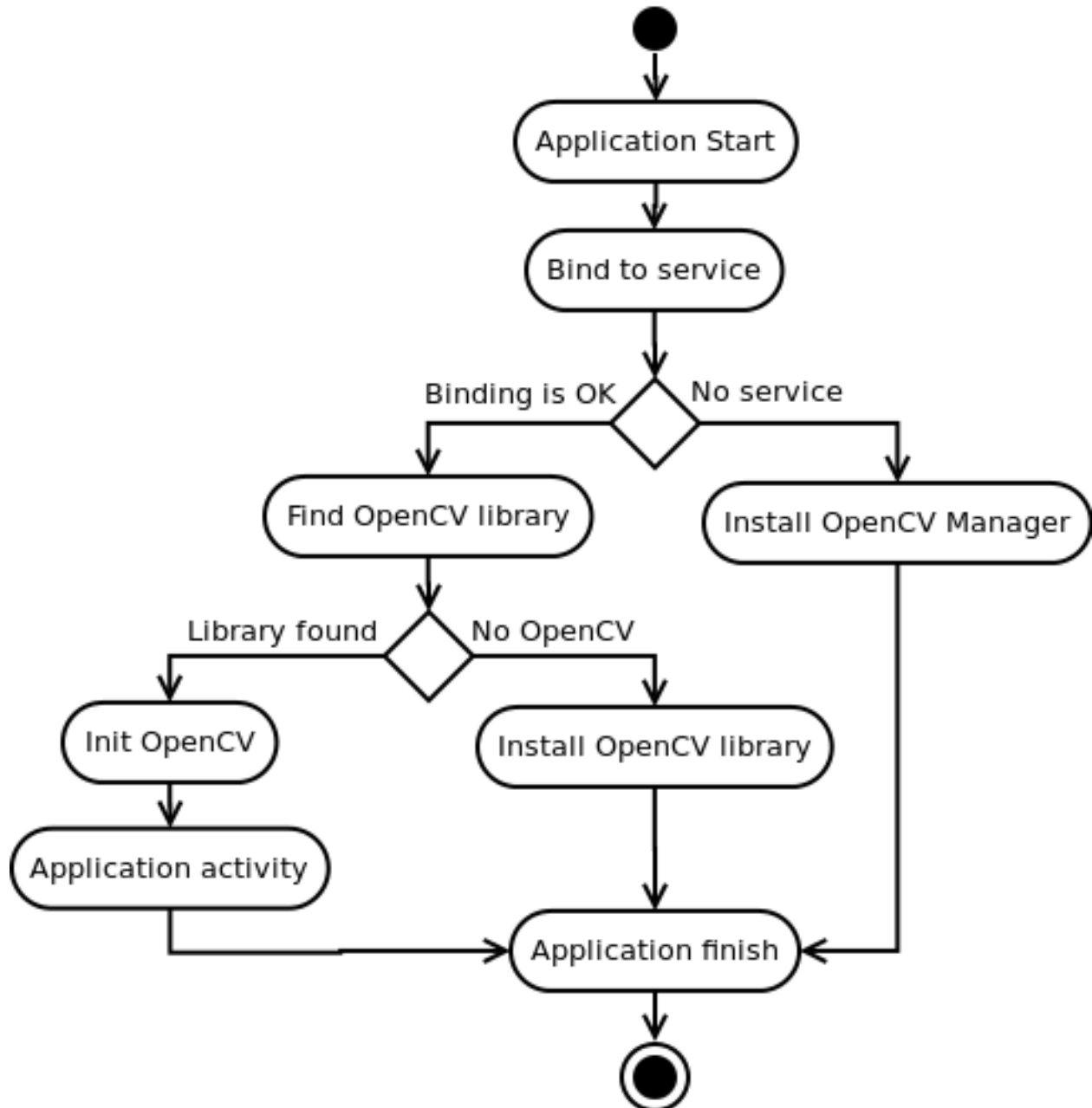
Contents:

## 1.1 Introduction

OpenCV Manager is an Android service targeted to manage OpenCV library binaries on end users devices. It allows sharing the OpenCV dynamic libraries of different versions between applications on the same device. The Manager provides the following benefits:

1. Less memory usage. All apps use the same binaries from service and do not keep native libs inside themselves;
2. Hardware specific optimizations for all supported platforms;
3. Trusted OpenCV library source. All packages with OpenCV are published on Google Play service;
4. Regular updates and bug fixes;

## Usage model for target user



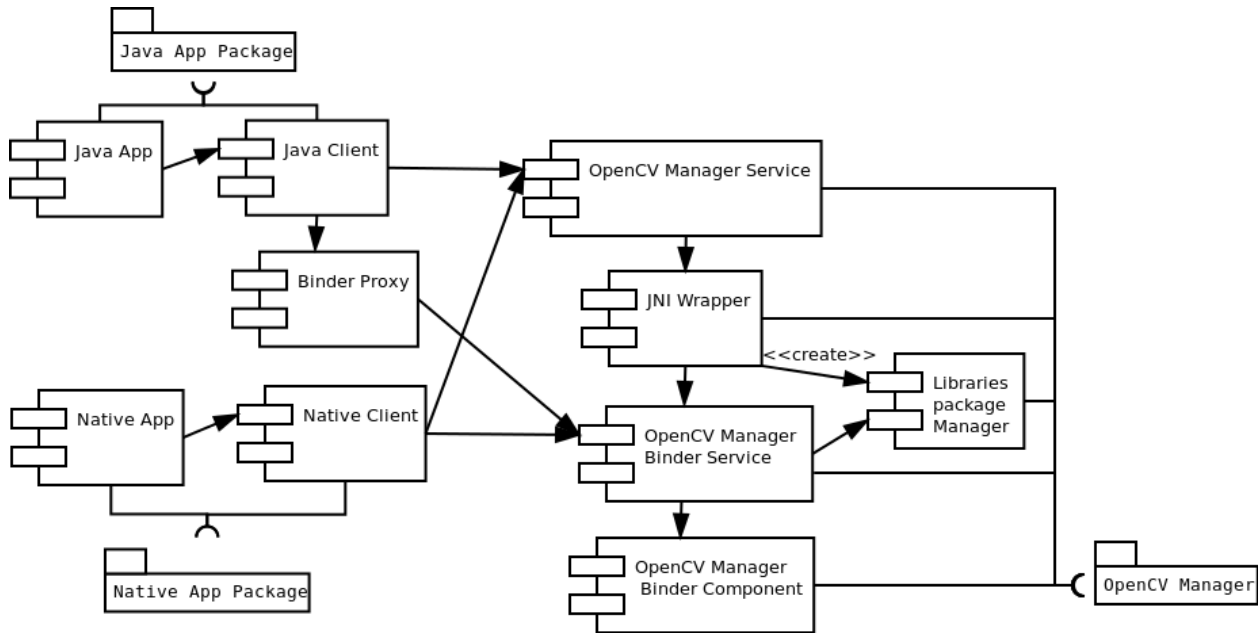
First OpenCV app:

1. Any OpenCV-dependent app is installed from Google Play marketplace or manually;
2. At the first launch, it suggests installing OpenCV Manager;
3. Then OpenCV Manager is downloaded and installed, using Google Play marketplace service.
4. When Manager has been started, the application suggests installing OpenCV library for the target device through Google Play marketplace if it is necessary;
5. After installation is finished, the app may be launched to perform common tasks.

Next OpenCV app:

1. Any OpenCV-dependent app is installed from Google Play marketplace or manually;
2. At the first launch, the app starts as usually;
3. If the selected version is not installed, OpenCV Manager suggests installing OpenCV library for the target device trough Google Play marketplace;
4. After installation is finished, the app may be launched to perform common tasks.

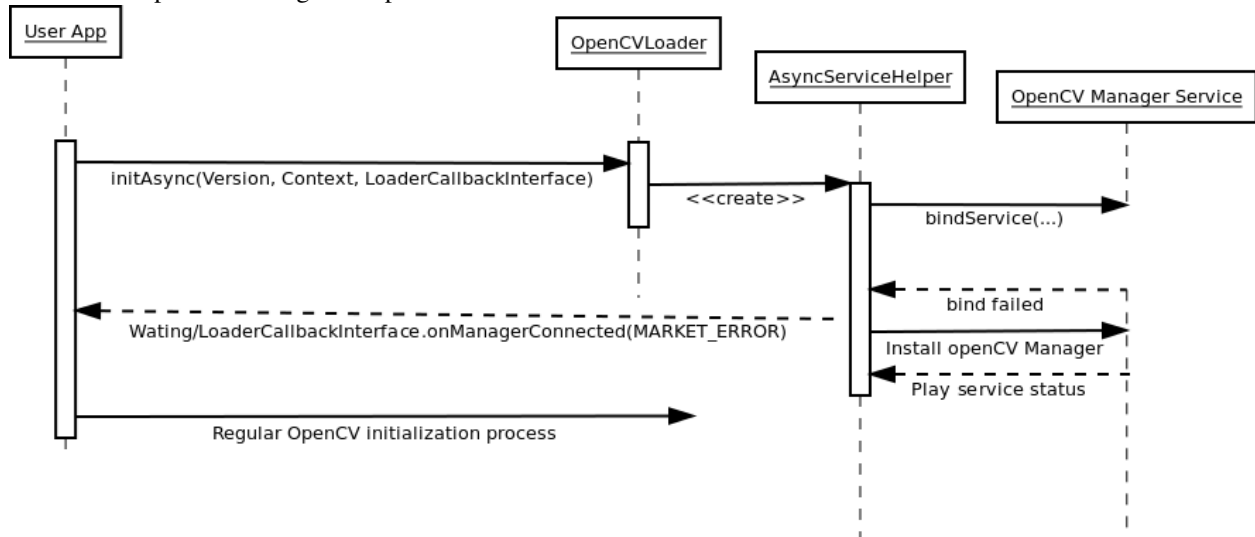
## OpenCV Manager structure



## 1.2 Manager Workflow

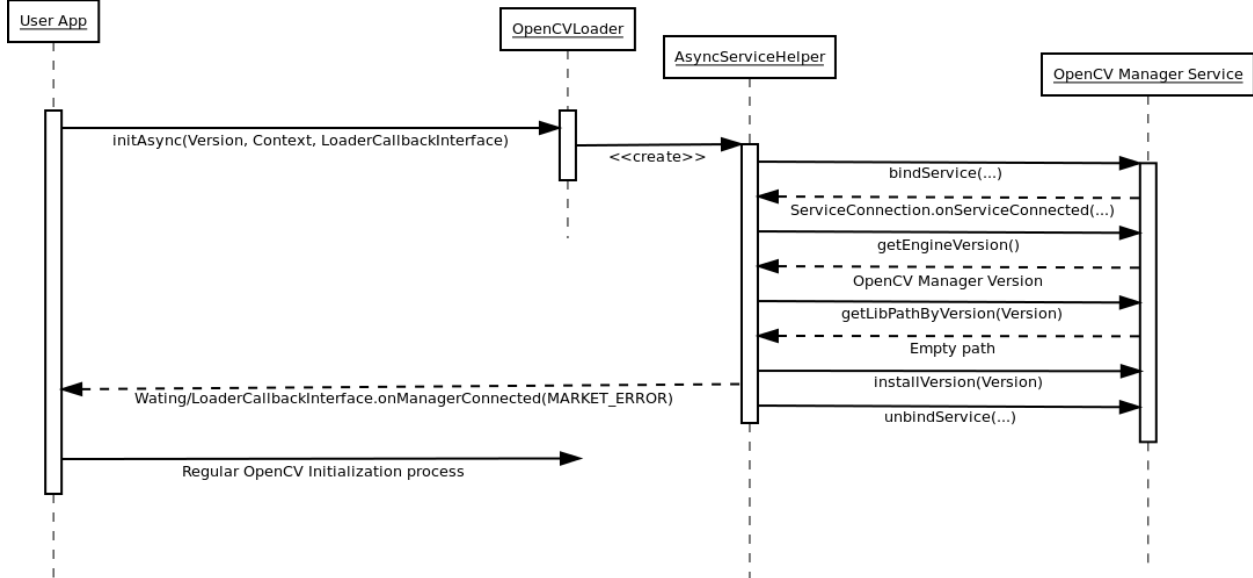
### First application start

There is no OpenCV Manager or OpenCV libraries:

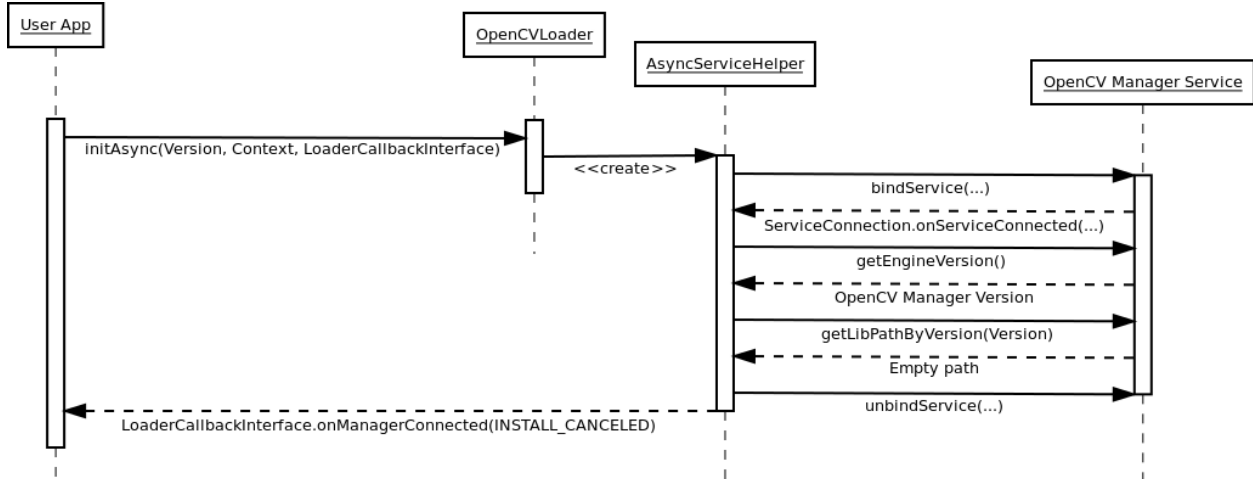


## Additional library package installation

There is an OpenCV Manager service, but there is no appropriate OpenCV library. If OpenCV library installation has been approved:



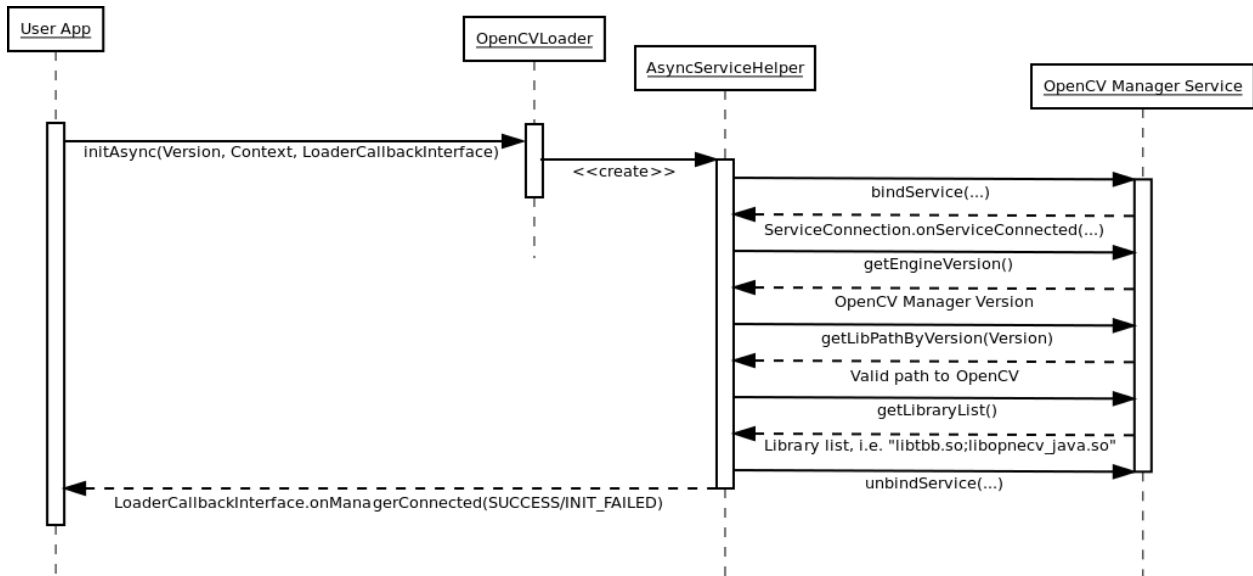
If OpenCV library installation has been canceled:



## Regular application start

OpenCV Manager and OpenCV library has been already installed.





## 1.3 Java OpenCV Loader

### class `OpenCVLoader`

Helper class provides common initialization methods for OpenCV library.

### boolean `initDebug()`

#### static boolean `initDebug()`

Loads and initializes OpenCV library from within current application package. Roughly it is analog of `system.loadLibrary("opencv_java")`.

**Return type** boolean;

**Returns** returns true if initialization of OpenCV was successful.

---

**Note:** This method is deprecated for production code. It is designed for experimental and local development purposes only. If you want to publish your app use approach with async initialization.

---

### boolean `initAsync()`

#### static boolean `initAsync(String Version, Context AppContext, LoaderCallbackInterface Callback)`

Loads and initializes OpenCV library using OpenCV Manager service.

#### Parameters

**Version** – OpenCV Library version.

**AppContext** – application context for connecting to the service.

**Callback** – object, that implements `LoaderCallbackInterface` for handling connection status (see `BaseLoaderCallback`).

**Return type** boolean;

**Returns** returns true if initialization of OpenCV starts successfully.

## OpenCV version constants

### **OPENCV\_VERSION\_2\_4\_2**

OpenCV Library version 2.4.2

### **OPENCV\_VERSION\_2\_4\_3**

OpenCV Library version 2.4.3

## Other constants

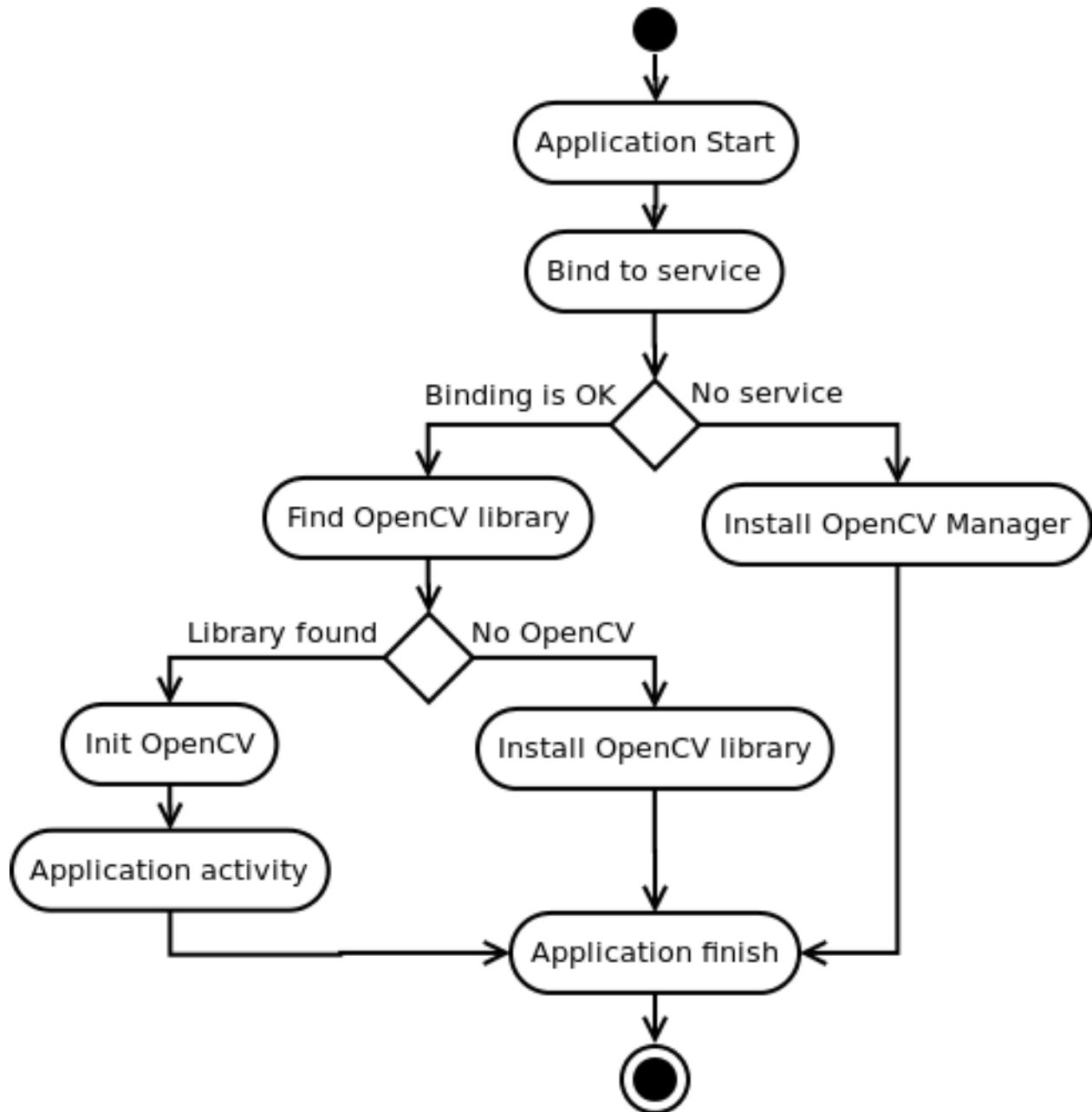
### **OPEN\_CV\_SERVICE\_URL**

Url for OpenCV Manager on Google Play (Android Market)

## 1.4 Base Loader Callback Interface implementation

### **class BaseLoaderCallback**

Basic implementation of LoaderCallbackInterface. Logic of this implementation is well-described by the following scheme:



## Using in Java Activity

There is a very base code snippet implementing the async initialization with BaseLoaderCallback. See the “15-puzzle” OpenCV sample for details.

```

1 public class MyActivity extends Activity implements HelperCallbackInterface
2 {
3     private BaseLoaderCallback mOpenCVCallBack = new BaseLoaderCallback(this) {
4         @Override
5         public void onManagerConnected(int status) {
6             switch (status) {
7                 case LoaderCallbackInterface.SUCCESS:

```

```
8     {
9         Log.i(TAG, "OpenCV loaded successfully");
10        // Create and set View
11        mView = new puzzle15View(mAppContext);
12        setContentView(mView);
13    } break;
14    default:
15    {
16        super.onManagerConnected(status);
17    } break;
18    }
19 }
20 };
21
22 /** Call on every application resume */
23 @Override
24 protected void onResume()
25 {
26     Log.i(TAG, "called onResume");
27     super.onResume();
28
29     Log.i(TAG, "Trying to load OpenCV library");
30     if (!OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_2_4_3, this, mOpenCVCallback))
31     {
32         Log.e(TAG, "Cannot connect to OpenCV Manager");
33     }
34 }
```

## Using in Service

Default BaseLoaderCallback implementation treat application context as Activity and calls Activity.finish() method to exit in case of initialization failure. To override this behavior you need to override finish() method of BaseLoaderCallback class and implement your own finalization method.

## 1.5 Loader Callback Interface

### class LoaderCallbackInterface

Interface for a callback object in case of asynchronous initialization of OpenCV.

### void onManagerConnected()

### void onManagerConnected(int status)

Callback method that is called after OpenCV Library initialization.

#### Parameters

**status** – status of initialization (see Initialization Status Constants).

### void onPackageInstall()

### void onPackageInstall(InstallCallbackInterface Callback)

Callback method that is called in case when package installation is needed.

### Parameters

**callback** – answer object with approve and cancel methods and package description.

## Initialization status constants

### SUCCESS

OpenCV initialization finished successfully

### MARKET\_ERROR

Google Play (Android Market) cannot be invoked

### INSTALL\_CANCELED

OpenCV library installation was canceled by user

### INCOMPATIBLE\_MANAGER\_VERSION

Version of OpenCV Manager Service is incompatible with this app. Service update is needed

### INIT\_FAILED

OpenCV library initialization failed

## 1.6 Install Callback Interface

### class **InstallCallbackInterface**

Callback interface for package installation or update.

### **String** **getPackageName()**

#### **String** **getPackageName()**

Get name of a package to be installed.

**Return type** string;

**Returns** returns package name, i.e. “OpenCV Manager Service” or “OpenCV library”.

### **void** **install()**

#### **void** **install()**

Installation of package has been approved.

### **void** **cancel()**

#### **void** **cancel()**

Installation if package has been canceled.

### **void** **wait\_install()**

#### **void** **wait\_install()**

Wait for package installation.



# JAVA API

[Java API reference external link \(JavaDoc\)](#)