



Certified Cordova Mobile  
Application Developer  
VS-1124

**V-Skills Certifications**

**A Government of India  
&  
Government of NCT Delhi Initiative**

**V-Skills**



### 1. OVERVIEW

---

Apache Cordova is an open-source mobile development framework. It allows you to use standard web technologies such as HTML5, CSS3, and JavaScript for cross-platform development, avoiding each mobile platform's native development language. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's sensors, data, and network status.

Apache Cordova graduated in October 2012 as a top level project within the Apache Software Foundation (ASF). Through the ASF, future Cordova development will ensure open stewardship of the project. It will always remain free and open source under the Apache License, Version 2.0. Visit [cordova.apache.org](http://cordova.apache.org) for more information.

Use Apache Cordova if you are:

- ✓ A mobile developer and want to extend an application across more than one platform, without having to re-implement it with each platform's language and tool set.
- ✓ A web developer and want to deploy a web app that's packaged for distribution in various app store portals.
- ✓ A mobile developer interested in mixing native application components with a WebView (special browser window) that can access device-level APIs, or if you want to develop a plugin interface between native and WebView components.

#### 1.1. Basic Components

Apache Cordova applications rely on a common `config.xml` file that provides information about the app and specifies parameters affecting how it works, such as whether it responds to orientation shifts. This file adheres to the W3C's Packaged Web App, or widget, specification.

The application itself is implemented as a web page, by default a local file named `index.html`, that references whatever CSS, JavaScript, images, media files, or other resources are necessary for it to run. The app executes as a WebView within the native application wrapper, which you distribute to app stores.

The Cordova-enabled WebView may provide the application with its entire user interface. On some platforms, it can also be a component within a larger, hybrid application that mixes the WebView with native application components.

A plugin interface is available for Cordova and native components to communicate with each other. This enables you to invoke native code from JavaScript. Ideally, the JavaScript APIs to that native code are consistent across multiple device platforms. As of version 3.0, plugins provide bindings to standard device APIs. Third-party plugins provide additional bindings to features not necessarily available on all platforms. You can find these third-party plugins in the plugin registry and use them in your application. You can also develop your own plugins, as described in the Plugin Development Guide. Plugins may be necessary, for example, to communicate between Cordova and custom native components.

As of version 3.0, when you create a Cordova project it does not have any plugins present. This is the new default behavior. Any plugins you desire, even the core plugins, must be explicitly added.

Cordova does not provide any UI widgets or MV frameworks. Cordova provides only the runtime in which those can execute. If you wish to use UI widgets and/or an MV framework, you will need to select those and include them in your application yourself as third-party material.

### 1.2. Development Paths

As of version 3.0, you can use two basic workflows to create a mobile app. While you can often use either workflow to accomplish the same task, they each offer advantages:

#### **Cross-platform (CLI) workflow**

Use this workflow if you want your app to run on as many different mobile operating systems as possible, with little need for platform-specific development. This workflow centers around the cordova utility, otherwise known as the Cordova CLI, that was introduced with Cordova 3.0. The CLI is a high-level tool that allows you to build projects for many platforms at once, abstracting away much of the functionality of lower-level shell scripts. The CLI copies a common set of web assets into subdirectories for each mobile platform, makes any necessary configuration changes for each, runs build scripts to generate application binaries. The CLI also provides a common interface to apply plugins to your app. Unless you have a need for the platform-centered workflow, the cross-platform workflow is recommended.

#### **Platform-centered workflow**

Use this workflow if you want to focus on building an app for a single platform and need to be able to modify it at a lower level. You need to use this approach, for example, if you want your app to mix custom native components with web-based Cordova components, as discussed in Embedding WebViews. As a rule of thumb, use this workflow if you need to modify the project within the SDK. This workflow relies on a set of lower-level shell scripts that are tailored for each supported platform, and a separate Plugman utility that allows you to apply plugins. While you can use this workflow to build cross-platform apps, it is generally more difficult because the lack of a higher-level tool means separate build cycles and plugin modifications for each platform. Still, this workflow allows you greater access to development options provided by each SDK, and is essential for complex hybrid apps.

When first starting out, it may be easiest to use the cross-platform workflow to create an app, as described in The Command-line Interface. You then have the option to switch to a platform-centered workflow if you need the greater control the SDK provides. Lower-level shell utilities are available at [cordova.apache.org](http://cordova.apache.org) in a separate distribution than the CLI. For projects initially generated by the CLI, these shell tools are also available in the project's various `platforms/*/cordova` directories.

Once you switch from the CLI-based workflow to one centered around the platform-specific SDKs and shell tools, you can't go back. The CLI maintains a common set of cross-platform source code, which on each build it uses to write over platform-specific source code. To preserve any modifications you make to the platform-specific assets, you need to switch to the platform-centered

shell tools, which ignore the cross-platform source code, and instead relies on the platform-specific source code.

### 1.3. Installing Cordova

The installation of Cordova will differ depending on the workflow above you choose:

- ✓ Cross-platform workflow: see The Command-Line Interface.
- ✓ Platform-centered workflow: see the Platform Guides.

After installing Cordova, it is recommended that you review the Platform Guides for the mobile platforms that you will be developing for. It is also recommended that you also review the Privacy Guide, Security Guide, and Next Steps. For accessing native function on a device from JavaScript, refer to the Plugin APIs. And refer to the other included guides as necessary.

The following shows the set of development tools and device APIs available for each mobile platform. The device APIs listed here are provided by the core plugins, additional APIs are available via third-party plugins. Column headers display the CLI's shorthand names.

### 1.4. Platform Support

Apache Cordova currently supports the following platforms

- ✓ Android (Google)—<http://developer.android.com/index.html>
- ✓ bada (Samsung)—<http://developer.bada.com>
- ✓ BlackBerry 10 (BlackBerry) —<https://developer.blackberry.com/>
- ✓ iOS (Apple) —<https://developer.apple.com/devcenter/ios/index.action>
- ✓ Firefox OS—[https://developer.mozilla.org/en-US/docs/Mozilla/Firefox\\_OS](https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS)
- ✓ Tizen (originally Samsung, now the Linux Foundation)—<https://developer.tizen.org>
- ✓ Windows Phone 7 and Windows Phone 8 (Microsoft) — <http://developer.windowsphone.com/en-us>
- ✓ Windows 8 (Microsoft)—<http://msdn.microsoft.com>

Details are listed below

	amazon-fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.0, 8.1, Phone 8.1)	tizen
<b>cordova CLI</b>	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓	X
<b>Embedded WebView</b>	✓ (see details)	✓ (see details)	X	X	✓ (see details)	✓	X	X	X
<b>Plug-in Interface</b>	✓ (see details)	✓ (see details)	✓ (see details)	X	✓ (see details)	✓	✓ (see details)	✓	X

	Platform APIs									
Accelerometer*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BatteryStatus*	✓	✓	✓	✓	✓	✗	✓	✓	✓ Windows Phone 8.1 only	✓
Camera*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Compass*	✓	✓	✓	✗	✓ (3GS+)	✓	✓	✓	✓	✓
Connection*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Contacts*	✓	✓	✓	✓	✓	✓	✓	✓	partially	✗
Device*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
File*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
File Transfer*	✓	✓	✓ Do not support onprogress nor abort	✗	✓	✗	✓ Do not support onprogress nor abort	✓ Do not support onprogress nor abort	✓ Do not support onprogress nor abort	✗
Geolocation*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalization*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
InAppBrowser*	✓	✓	✓	✗	✓	✓	✓	✓	uses iframe	✗
Media*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Notification*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Splashscreen*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Status Bar*	✗	✓	✗	✗	✓	✗	✓	✓	✓ Windows Phone 8.1 only	✗
Storage	✓	✓	✓	✗	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓	✓
Vibration*	✓	✓	✓	✓	✓	✗	✓	✓	✓ * Windows Phone 8.1 only	✗

### 1.5. Cordova Applications

Cordova applications are web applications running inside of a client- side native application container. Therefore, web applications running within a Cordova application leverage an HTML5 application structure rather than that of a traditional server- based web application.

Mobile devices need a slightly different approach. Web 1.0 and 2.0 technologies work great on smartphones, but Web 1.0 apps caused a lot of data to be transmitted between server and device, and Web 2.0 apps were cooler but still required constant network connectivity to operate.

With HTML5, web applications can make use of new capabilities that allow an application to operate more efficiently on a mobile device (or devices with limited connectivity), and they can use a client-side database to store application data. This functionality makes it easier for mobile devices to operate as they go in and out of wireless coverage. Additionally, HTML5 supports the addition of a manifest file that lists all of the files that comprise the web application. When the web application's index file loads, the browser reads the manifest file, retrieves all of the files listed in the manifest, and downloads them to the client device. If a mobile device were to lose network

connectivity, then as long as the files listed in the manifest were available on - device, the application could continue working—using any data that might be stored locally.

When a Cordova application launches, it loads the web application's startup page (typically `index.html`) and associated content (CSS files, JavaScript files) before passing control to the web app. In order for this method to work, the resources the app needs to start have to be located within the container.

Some Cordova developers load as little as possible within the container and, immediately after startup, run off to a server to get the “real” content for the application; I see their questions on the support forums all the time. This approach works, but it's not the best experience for users and may cause you problems with app store submissions—some smartphone platforms (Apple iOS, for example) don't like it when your app doesn't contain content and is merely a shell for a web application being hosted by a web server

### IDE

Various IDEs are available for coding. Adobe, offers a free, open source code editor called Brackets (<http://brackets.io> ) and it provides a nice, clean interface for coding your web applications. The open source Aptana studio ([www.aptana.com](http://www.aptana.com)) is another option, it's an open source Eclipse - based IDE tailored for web development. It's lighter - weight than Eclipse.

### Packaging and Deployment

Web application, whether it uses any of the Cordova APIs or not, it has to be packaged into a native application that will run on - device. Platforms supported by the Cordova project have proprietary tools for packaging or building native applications. To build a Cordova application for each supported mobile platform, the application's web content (the HTML, CSS, JavaScript, and other files that comprise the application) must be added to an application project appropriate for each mobile platform, then built using the platform's proprietary tools.

Some supported platform tools run only on certain desktop operating systems, as

- ✓ The Android SDK runs on Linux, Microsoft Windows, and Macintosh OS X .
- ✓ The BlackBerry SDKs (they have several) run on Microsoft Windows and Macintosh OS X .
- ✓ The iOS SDK runs only on Macintosh OS X (no surprise there).
- ✓ The Windows Phone SDK runs only on Microsoft Windows (no surprise there either).

### Certifications

#### ► Accounting, Banking & Finance

- Certified AML- KYC Compliance Officer
- Certified Business Accountant
- Certified Commercial Banker
- Certified Equity Research Analyst
- Certified Foreign Exchange Professional
- Certified Hedge Fund Manager
- Certified Merger and Acquisition Analyst
- Certified Tally 9.0 Professional
- Certified Treasury Markets Professional
- Certified Wealth Manager

#### ► Foreign Trade

- Certified Export Import (Foreign Trade) Professional

#### ► Hospitality

- Certified Restaurant Team Member (Hospitality)

#### ► Human Resources

- Certified HR Compensation Manager
- Certified HR Staffing Manager
- Certified Human Resources Manager
- Certified Performance Appraisal Manager

#### ► Logistics & Supply Chain Management

- Certified International Logistics Professional
- Certified Logistics & SCM Professional
- Certified Purchase Manager

#### ► Law

- Certified IPR & Legal Manager

#### ► Life Skills

- Certified Business Communication Specialist
- Certified Public Relations Officer

#### ► Media

- Certified Advertising Manager
- Certified Advertising Sales Professional

#### ► Office Skills

- Certified Data Entry Operator
- Certified Office Administrator

#### ► Project Management

- Certified Project Management Professional

#### ► Real Estate

- Certified Real Estate Consultant

#### ► Information Technology

- Certified Android Apps Developer
- Certified ASP.NET Programmer
- Certified Basic Network Support Professional
- Certified Business Intelligence Professional
- Certified C# Professional
- Certified CAD Professional
- Certified Cloud Computing Professional
- Certified Computer Fundamentals (MS Office) Professional
- Certified Core Java Developer
- Certified CSS Designer
- Certified Data Mining and Warehousing Professional
- Certified DHTML & Javascript Developer
- Certified Django Developer
- Certified DTP operator
- Certified E-commerce Professional
- Certified E-Governance Professional
- Certified Enterprise Applications Integration Specialist (Biztalk)
- Certified Ethical Hacking and Security Professional
- Certified Facebook Apps Developer
- Certified Grid Computing Professional
- Certified Hadoop and Mapreduce Professional
- Certified HTML Designer
- Certified HTML5 Developer
- Certified iPhone Apps Developer
- Certified IT Support Professional
- Certified J2ME Programmer
- Certified Joomla Developer
- Certified Linux Administrator
- Certified Magento Professional
- Certified MySQL DB Administrator
- Certified Network Security Professional
- Certified Open Source CMS (Drupal) Professional
- Certified PHP Professional
- Certified PL/SQL Developer
- Certified Python Professional
- Certified Router Support Professional
- Certified Selenium Professional
- Certified SEO Professional
- Certified Software Quality Assurance Professional
- Certified Software Security Professional
- Certified Software Testing Professional
- Certified SQL Server 2008 Programmer
- Certified WiMax(4G) Professional
- Certified Wordpress Developer
- Certified XML Developer

#### ► Sales, BPO

- Certified Telesales Executive

Contact us at:

**V-Skills**

011-473 44 723 or [info@vskills.in](mailto:info@vskills.in)

**Intelligent Communication Systems India Limited**

DSIIDC Administrative Building,  
Okhla Industrial Estate-III, New Delhi-110020

[www.vskills.in](http://www.vskills.in)