Installation: SAP Mobile Platform SDK for Mac OS

SAP Mobile Platform 3.0 and SP01

Mac OS

DOCUMENT ID: DC01996-01-0301-02
LAST REVISED: December 2013

Contents

Installing SAP Mobile Platform SDK

SAP® Mobile Platform SDK works closely with SAP Mobile Platform Server, but its installation is independent.

1. *Preparing for Installation*

   Ensure that the host on which you are installing SAP Mobile Platform SDK meets the system requirements and is ready for you to begin the installation.
2. *Installing SDK Component Files*

   You can either obtain physical media and copy the SAP Mobile Platform SDK component files, or you can download a Web archive and extract the files.

Preparing for Installation

Ensure that the host on which you are installing SAP Mobile Platform SDK meets the system requirements and is ready for you to begin the installation.

Verify that the target system meets the system requirements for SAP Mobile Platform SDK.

See the SAP Product Availability Matrix (PAM) *http://service.sap.com/pam*. Click the **Mobile** link at the top of the page. Scroll to find the appropriate product and version in the product list. Click the **Technical Release Information** tab for the list of supported operating systems and databases. In the **General Information** tab, click **Additional Release Information** in the Related Links section for additional system requirements, supported development environments, devices, and third-party compatibility.

For information on specific requirements for developing different types for different devices, see the list of SAP Notes in *Supported Versions for SAP Mobile Platform SDK* in *New Features*.

Installing SDK Component Files

You can either obtain physical media and copy the SAP Mobile Platform SDK component files, or you can download a Web archive and extract the files.

1. Use one of these methods:

   - Insert the physical installation media.
   - Download from the software distribution center on SAP Service Marketplace:
     a. Go to *http://service.sap.com/swdc*.

   b. Under "A-Z Index" click **Installations and Upgrades**.
   c. Click **M**.
   d. Click **SAP MOBILE PLATFORM SDK**.
   e. Click the link for the current version.
   f. Download the package for the SAP Mobile Platform SDK installer.
   g. Extract the entire contents of the installer `.zip` file to a temporary directory on a local disk on the target host.

      Use a short path, preferably a folder directly below the root directory, such as . The path can include only ASCII alphanumeric characters, underscore (_), hyphen (-), and period (.). Two consecutive period characters are not allowed.

2. Install the SAP Mobile Platform SDK component files that you wish to use.

   Note: The AgentryToolkit and SMSToolkit files are Windows-specific. Do not install them for use in Mac OS development.

   - From physical media, select and copy the top level folders for the components you wish to use to a directory that is convenient for your development environment.
   - From a Web archive, select and extract the top level folders for the components you wish to use to a directory that is convenient for your development environment.

Next

Install support package 01: *Upgrading to Support Package 01* on page 3.

Upgrading to Support Package 01

Upgrade SAP Mobile Platform SDK 3.0 with SP01.

Installing SDK SP01 Component Files

You can either obtain physical media and copy the SAP Mobile Platform SDK component files, or you can download a Web archive and extract the files.

1. Use one of these methods:
   - Insert the physical installation media.
   - Download from the software distribution center on SAP Service Marketplace:
     a. Go to *http://service.sap.com/swdc*.
     b. Under "A-Z Index" click **Installations and Upgrades**.
     c. Click **M**.
     d. Click **SAP MOBILE PLATFORM SDK**.
     e. Click the link for the current version.
     f. Download the package for the SAP Mobile Platform SDK installer.
     g. Extract the entire contents of the installer `.zip` file to a temporary directory on a local disk on the target host.

        Use a short path, preferably a folder directly below the root directory, such as . The path can include only ASCII alphanumeric characters, underscore (_), hyphen (-), and period (.). Two consecutive period characters are not allowed.
2. Install the support package upgrades for the SAP Mobile Platform SDK component files that you wish to use.
   - From physical media, select and copy the top level folders for the components you wish to use to the same directory to which you copied the top level folders for the 3.0 version of the components, overwriting the previous folders and files.
   - From a Web archive, select and extract the top level folders for the components you wish to use to the same directory to which you copied the top level folders for the 3.0 version of the components, overwriting the previous folders and files.

Note: You may add components that are not currently installed, but do not add AgentryToolkit or SMSToolkit – these files are Windows-specific and cannot be used in Mac OS development.

Next

If you are upgrading the Kapsel component, and you have any Cordova projects created with SAP Mobile Platform SDK version 3.0:

1. Continue with *Upgrading Kapsel Applications* on page 4.
2. Then perform *Postinstallation Tasks* on page 9.

If you have no Cordova projects created with SAP Mobile Platform SDK version 3.0, perform *Postinstallation Tasks* on page 9.

Upgrading Kapsel Applications

Follow these steps to upgrade your Kapsel plugins to use Cordova 3.1.

Prerequisites

- Perform the upgrade to Support Package 01
- Close any open Cordova projects
- Create a backup of your Cordova 3.0 projects

Task

Kapsel plugins are compiled for a specific version of the Cordova framework. As subsequent versions of Cordova are released, the Kapsel plugins must be tested, updated (if necessary), and built for release, against the updated Cordova version. This means that you cannot immediately use new Cordova versions for Kapsel applications—you must wait for the compatible version of Kapsel to be released before building Kapsel applications using the updated version of Cordova.

For example, the initial version of Kapsel supported Cordova 3.0 and Cordova command line interface 3.0.9, so you could use only Cordova 3.0 with the initial version of Kapsel. SAP Mobile Platform SP01 adds support for Cordova 3.1.

Note: Do not use a version of Cordova that is not compatible with the current version of the Kapsel SDK. If you upgrade your project to an incompatible version of Cordova, the Kapsel plugins may still install, but the application will not build properly and may not run on a mobile device or emulator.

All of the commands in this procedure are performed in a terminal window on Mac, or a Windows command prompt window, in the Cordova project directory. If you have more than one Cordova project to upgrade, you must perform the same steps in each Cordova project directory.

1. To determine which version of Cordova the project was created with, navigate to the directory that contains your Cordova project and enter:

   ```
   cordova platform
   ```

   For example:

```
C:\Kapsel_Projects\StorageDemo>cordova platform
```

You see something similar to:

```
Installed platforms: android 3.0.9
```

This example shows that Cordova 3.0.9 is installed for the Android platform.

2. Determine which version of the Cordova command line interface is installed by entering:

   ```
   cordova -v
   ```

3. See if there is a newer version of Cordova by entering:

   ```
   npm info cordova
   ```

   You see output similar to this:

   ```
   npm http GET https://registry.npmjs.org/cordova
   npm http 304 https://registry.npmjs.org/cordova

   { name: 'cordova',
     description: 'Cordova command line interface tool',
     'dist-tags': { latest: '3.1.0-0.2.0' },
     versions:
      [ '0.0.1',
        '0.0.2',
   ...
       '3.0.6',
       '3.0.7',
       '3.0.8',
       '3.0.9',
       '3.0.10',
       '3.1.0-0.1.0',
       '3.1.0-0.2.0' ],
   ```

4. Upgrade to the specified Cordova version by entering:

   On Windows: `npm install -g cordova@<current_cordova_version>-<upgrade_cordova_version>`

   On Mac: `sudo npm install -g cordova@<current_cordova_version>-<upgrade_cordova_version>`

   For example, on Windows, to upgrade from Cordova 3.1.0 to version 0.2.0, enter:

   ```
   npm install -g cordova@3.1.0-0.2.0
   ```

5. Upgrade the Cordova project by entering:

   For Android `cordova -d platform update android`

   For iOS `cordova -d platform update ios`

   The -d flag indicates debug output and is optional.

   See *http://cordova.apache.org/docs/en/3.1.0/guide_platforms_android_upgrading.md.html#Upgrading%20Android* for Android and *http://cordova.apache.org/docs/en/3.1.0/*

guide_platforms_ios_upgrading.md.html#Upgrading%20iOS for iOS for more information.

6. To see what plugins you have installed, navigate to the Kapsel project folder, and enter:

   ```
   cordova plugins
   ```

   You see a list of the plugins included with the application, for example:

   ```
    ['com.sap.mp.cordova.plugins.appupdate',
   'com.sap.mp.cordova.plugins.authproxy',
   'com.sap.mp.cordova.plugins.corelibs',
   'com.sap.mp.cordova.plugins.encryptedstorage',
   'com.sap.mp.cordova.plugins.i18n',
   'com.sap.mp.cordova.plugins.logger',
   'com.sap.mp.cordova.plugins.logon',
   'com.sap.mp.cordova.plugins.push',
   'com.sap.mp.cordova.plugins.settings',
   'org.apache.cordova.console',
   'org.apache.cordova.device',
   'org.apache.cordova.device-orientation',
   'org.apache.cordova.dialogs',
   'org.apache.cordova.inappbrowser' ]
   ```

   In the above example, you see that both Kapsel and Cordova Core plugins appear. This is because some of the Kapsel plugins rely on some of the Cordova Core APIs.

7. Uninstall each of the Kapsel plugins you have installed by entering:

   ```
   cordova plugin remove <plugin_id>
   ```

   Where *<plugin_id>* refers to the ID for the particular plugin. For example, to uninstall the Logon plugin, you would enter:

   ```
   cordova plugin remove com.sap.mp.cordova.plugins.logon
   ```

   Repeat this step for each Kapsel plugin, changing the plugin ID as needed, according to the plugin you are removing.

   Note: Since some of the Kapsel plugins depend on several of the Cordova plugins and those Cordova plugins may have been updated with each new release of Cordova, the dependent Cordova plugins must also be removed from the project before you reinstall the Kapsel plugins. Beginning with Cordova command line interface 3.1, dependent plugins should be removed when all plugins that rely upon them are removed, so when you complete these steps, both the Kapsel and associated, dependent Cordova Core API plugins should be removed from the project. However, testing has proven that this is not always the case. If, after removing all of the Kapsel plugins, you execute the **cordova plugins** command from a terminal window, you may see remnant plugins left in the project as shown in the example below.

8. Add the Kapsel plugins back to the project by entering:

   ```
   cordova plugin add <path_to_Kapsel_plugin>
   ```

   For example, to add the AppUpdate plugin, enter:

   On Windows:

```
cordova -d plugin add <SDK_HOME>\MobileSDK3\KapselSDK
\plugins\appupdate
```

On Mac:

```
cordova -d plugin add ~<SDK_HOME>/MobileSDK3/KapselSDK/
plugins/appupdate
```

Note: The path you enter to the Kapsel plugin must be the absolute path (not relative path).

9. Prepare the Cordova project by entering:

```
cordova -d prepare android
```

or

```
cordova -d prepare ios
```

10. Upgrade the Kapsel command line interface by entering:

```
SDK_HOME\MobileSDK3\KapselSDK\cli>npm uninstall -g kapsel
SDK_HOME\MobileSDK3\KapselSDK\cli>npm install -g
```

Postinstallation Tasks

Continue with the developer documentation for the type of development supported by the SAP Mobile Platform SDK components you have just installed.

Go to the *Setting Up the Development Environment* topic in the applicable developer documentation from the list below.

SDK Component	Developer Documentation
Native SDK	*Native OData App Development*
Kapsel SDK	*Kapsel Development*

Uninstalling SAP Mobile Platform SDK

Uninstall SAP Mobile Platform SDK by manually removing installed files.

1. Back up any user-created files and log files you want to keep from the installation directories to another location.
2. Manually delete folders and files in the *SDK_HOME* tree.

Troubleshooting

Determine the cause of common problems and apply the recommended solution.

Issues Requiring Product Support

Your SAP support ID gives you access to enterprise-level incident support as part of your support plan on SAP Service Marketplace.

Product Support can help you resolve new undocumented incidents with software installation, start-up, and overall use, as well as providing diagnostic and troubleshooting assistance for known problems with a new or undocumented cause.

Product Support Engagement Requirements

If you use SAP Service Marketplace to engage with Product Support, you must meet certain requirements.

Service Marketplace Case Creation Requirements
Be prepared to provide:

- A valid installation number for SAP Mobile Platform
- A valid service contract with SAP
- A valid system ID (S-User ID)
- An enabled NetViewer connection.

SAP Mobile Platform Incident Requirements

- Configure your logs to an appropriate level for your issue. Product Support requires details from one or more of the system logs.
- Capture these basic incident details to help Product Support analyze the problem, and determine any next steps:
  - Environment summary: product version, back end, client type (device and OS), proxy connections. These details help isolate component that is causing the failure. If you have an architecture diagram, share it with SAP.
  - Problem description: what were the actions preceded the incident. Capture all details that allow Product Support to reproduce the issue.
- Locate the server version in the SMP_HOME`\Server\version.properties` file.

Creating an Incident on SAP Service Marketplace

If you cannot resolve problems with the troubleshooting documentation for SAP Mobile Platform, go to SAP Service Marketplace for additional help.

Use SAP Service Marketplace to create an incident message for Product Support. Keywords from this message return related articles from the Knowledge Base. Before you submit a message, review these articles to see if they resolve your problem.

1. Go to *http://service.sap.com/message*.
2. Create a message using the wizard.

   Note: You must know the component ID for SAP Mobile Platform to return the correct scope of Knowledge Base Articles and to correctly route the message to Product Support. On-premise installations of SAP Mobile Platform use a different ID than cloud instances. See Knowledge Base Article *1915061- How to Choose a Component for SAP Mobile Platform 3.x in Service Marketplace* .
3. Once the message is processed, you receive an e-mail notification of the solution.

Installation Directories

Find the component files necessary to develop different types of applications.

The following table shows the top-level subdirectories that are created when you extract the contents of the SAP Mobile Platform SDK `.zip` file. The directory names identify the type of application that the different components support.

These directories appear immediately below the directory into which you extract the contents of the SAP Mobile Platform SDK `.zip` file.

Note: In the root of the SAP Mobile Platform SDK `.zip` file is an icon file, `smp-sdk-in-icon.ico`. This is not used by any of the SDK components.

Table 1. SAP Mobile Platform SDK Installation Subdirectories

Directory	Description
`AgentryToolkit`	Agentry Editor, Agentry Test Environment, and support files. **Note:** These files are Windows-specific. Do not use them in Mac OS development.
`ClientHub`	Files for Client Hub applications, used for registering applications on iOS and Android devices.
`KapselSDK`	Kapsel framework files.
`NativeSDK`	Native application files. that support Mobile Application Framework (MAF) and OData Framework.
`SMSToolKit`	Short Message Service (SMS) files, that support SMS Builder. **Note:** These files are Windows-specific. Do not use them in Mac OS development.
ThirdParty	License terms of third-party components that are included in SAP Mobile Platform SDK.

Index