

SAP Mobile Platform 3.0 SP04
Document Version: 1.0 - 2014-06-25

Upgrade: SAP Mobile Platform SDK for Mac OS



Table of Contents

1	Upgrade: SAP Mobile Platform SDK for Mac OS.	3
2	Upgrading SAP Mobile Platform 3.0 SDK to SP04.	4
2.1	Acquiring the Support Package Installer.	4
2.2	Preparing to Upgrade.	5
2.3	Running the Support Package Installer.	5
2.3.1	Upgrading Kapsel Applications.	6
2.4	Verifying the Upgrade.	10
3	Troubleshooting.	11
3.1	Issues Requiring Product Support.	11
3.1.1	Product Support Engagement Requirements.	11
3.1.2	Creating an Incident on SAP Service Marketplace.	12
4	Postinstallation Tasks.	13
5	Important Disclaimers on Legal Aspects.	14

1 Upgrade: SAP Mobile Platform SDK for Mac OS

Upgrade SAP® Mobile Platform SDK to Support Package 04 to access new features.

Upgrading SAP Mobile Platform SDK includes verifying system requirements, acquiring and running the support package installer, and verifying the upgrade. You can also upgrade your Kapsel plug-ins to use a later version of Cordova.

Postinstallation Tasks directs you to documentation that explains how to develop applications with the components you installed.

Related Information

[Upgrading SAP Mobile Platform 3.0 SDK to SP04](#) [page 4]

Upgrade SAP Mobile Platform SDK 3.0 installations by running the Support Package 04 installer.

[Troubleshooting](#) [page 11]

Review information about common problems that arise in the SAP Mobile Platform SDK upgrade process.

[Postinstallation Tasks](#) [page 13]

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

2 Upgrading SAP Mobile Platform 3.0 SDK to SP04

Upgrade SAP Mobile Platform SDK 3.0 installations by running the Support Package 04 installer.

1. [Acquiring the Support Package Installer](#) [page 4]
The SAP Mobile Platform SDK SP04 installer is available as physical media and as a Web download.
2. [Preparing to Upgrade](#) [page 5]
Ensure that the host on which SAP Mobile Platform SDK is installed is ready for you to begin the upgrade.
3. [Running the Support Package Installer](#) [page 5]
Start the SAP Mobile Platform SDK SP04 installer, accept the current installation directory, select components to upgrade, and launch the upgrade process.
4. [Verifying the Upgrade](#) [page 10]
After you complete the upgrade, verify that it is successful.

2.1 Acquiring the Support Package Installer

The SAP Mobile Platform SDK SP04 installer is available as physical media and as a Web download.

Procedure

Use one of these methods:

- Insert the physical installation media.
- Download from the software distribution center on SAP Service Marketplace:
 1. Go to <http://service.sap.com/swdc>.
 2. In the left pane, under *Support Packages and Patches*, select *A-Z Index*.
 3. Click *M*.
 4. Select *SAP MOBILE PLATFORM SDK*.
 5. Click the link for the current version.
 6. Select the platform.
 7. Choose the file to download.
 8. Extract the contents of the .zip file to a temporary directory on a local disk on the target host.
The download .zip file contains two other .zip files, the SAP Mobile Platform SDK installer and the MBO Toolkit installer.
 9. Extract the contents of the SAP Mobile Platform SDK 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 `/Users/<user_name>/temp`. The path can include only ASCII alphanumeric characters, underscore (_), hyphen (-), and period (.). Two consecutive period characters are not allowed.

You need not extract the contents of the MBO Toolkit installer .zip file; the SAP Mobile Platform SDK installer unzips it automatically.

2.2 Preparing to Upgrade

Ensure that the host on which SAP Mobile Platform SDK is installed is ready for you to begin the upgrade.

Context

Procedure

1. 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.

2. Verify that Java for OS X, version 6 or higher, is installed on the target system.

2.3 Running the Support Package Installer

Start the SAP Mobile Platform SDK SP04 installer, accept the current installation directory, select components to upgrade, and launch the upgrade process.

Context

Note

The installation path you accept below is referred to as **<SDK_HOME>** in the rest of these instructions.

Procedure

1. Browse to the root directory of the SAP Mobile Platform installer and double-click the *setupMacOSX.jar* file.

Note

The installer displays an `Initializing wizard...` message while it verifies the integrity of the installation image. It may take a few minutes for the first installer panel to appear.

2. On the installer welcome page, click [Next](#).
3. On the end-user license agreement page, select your country, accept the terms of the license agreement and click [Next](#).
4. Accept the default installation or navigate to another location.
The SAP Mobile Platform SDK 3.0 SP04 support package installer for Mac OS does not detect earlier SDK installations. If you want the installer to replace the earlier SDK component files, you must navigate to the directory below which the folders for those components are located.
5. Select the specific components to upgrade.
If you select components that are not currently installed, the installer adds them. Any currently installed components that are not located in folders immediately below the installation directory you specified are not affected.
6. On the summary information panel, verify the installation features and click [Upgrade](#).
7. Click [Finish](#).

Next Steps

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](#) [page 6].
2. Then go on to [Verifying the Upgrade](#) [page 10].

If you have no Cordova projects created with SAP Mobile Platform SDK version 3.0, go on to [Verifying the Upgrade](#) [page 10].

2.3.1 Upgrading Kapsel Applications

Follow these steps to upgrade your Kapsel plugins to use a specific version of Cordova.

Prerequisites

- Perform the upgrade to Support Package 04
- Close any open Cordova projects
- Create a backup of your Cordova 3.0, Cordova 3.1, or Cordova 3.3 projects

Context

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 SDK SP01 adds support for Cordova 3.1; SP02 adds support for Cordova 3.3; and SP03 adds support for Cordova 3.4.

i 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.

Procedure

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:

```
SDK_HOME/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: '3.4.0-0.1.3' },
  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',
'3.3.1-0.1.2',
'3.4.0-0.1.3' ],
```

4. Upgrade to the specified Cordova version by entering:

On Windows: **npm install -g cordova@<version_to_upgrade_to>**

On Mac: **sudo npm install -g cordova@<version_to_upgrade_to>**

For example, on Windows, to upgrade from Cordova 3.1.0 to version 3.4.0, enter:

npm install -g cordova@3.4.0-0.1.3

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/3.3.0/guide_platforms_android_index.md.html#Android%20Platform%20Guide for Android and http://cordova.apache.org/docs/en/3.3.0/guide_platforms_ios_index.md.html#iOS%20Platform%20Guide 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.

i 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. For iOS, when removing plugins, common frameworks may be removed. To allow the upgraded project to be successfully built, you can either again add the frameworks to the Xcode project, or recreate the project.

In the Xcode project, you can add these frameworks:

- CFNetwork.framework
- SystemConfiguration.framework
- AudioToolbox.framework

Alternatively, you can recreate your project from scratch and copy over the `www` folder.

9. 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
```

i Note

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

10. Prepare the Cordova project by entering:

```
cordova -d prepare android
```

or

```
cordova -d prepare ios
```

11. 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
```

2.4 Verifying the Upgrade

After you complete the upgrade, verify that it is successful.

1. Use a text editor to open the latest `SMP SDKInstall<<date-time>>.log` file from the `<SMP_HOME>/InstallLogs` directory.
By default, SAP Mobile Platform SDK components are installed in the `<$HOME>/SAP/MobileSDK<XXX>` directory. In this guide, `<SDK_HOME>` represents the SAP Mobile Platform SDK installation directory, down to the `MobileSDK<XXX>` folder.
2. Search for "error"—no search results indicates that the installation completed successfully.

Next

Perform [Postinstallation Tasks](#) [page 13].

3 Troubleshooting

Review information about common problems that arise in the SAP Mobile Platform SDK upgrade process.

For information about contacting SAP Technical Support, see [Issues Requiring Product Support](#) [page 11].

3.1 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.

3.1.1 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.


3.1.2 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.

4 Postinstallation Tasks

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

Procedure

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

SDK Component	Developer Documentation
Native SDK	<i>Native OData App Development</i>
Kapsel SDK	<i>Kapsel Development</i>

5 Important Disclaimers on Legal Aspects

This document is for informational purposes only. Its content is subject to change without notice, and SAP does not warrant that it is error-free. SAP MAKES NO WARRANTIES, EXPRESS OR IMPLIED, OR OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP specifically disclaims any liability with respect to this document and no contractual obligations or commitments are formed either directly or indirectly by this document.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. Regarding link classification, see: <http://help.sap.com/disclaimer>





www.sap.com/contactsap

© 2014 SAP AG or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

