



**Mobile Data Models: Using Data
Orchestration Engine**

SAP Mobile Platform 2.3 SP03

DOCUMENT ID: DC01935-01-0233-01

LAST REVISED: September 2013

Copyright © 2013 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

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 in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Getting Started with DOE-Based Application	
Development	1
SAP Data Orchestration Engine (DOE)	1
DOE-Based Application Scenarios	2
Application Architecture	3
Developer Roles	4
SAP Data Structure Customization	4
Development Task Flow	7
Creating and Encapsulating BAPIs	8
Creating a Software Component Version	8
Modeling Data Objects	9
Modeling Distribution Logic	9
Creating, Generating, and Downloading the ESDMA	
Bundle	10
Converting the ESDMA Bundle into an SAP Mobile	
Platform Package	10
Deploying the SAP Mobile Platform Package	12
Developing Platform-Specific Device Code	13
Documentation Roadmap for SAP Mobile	
Platform	13
Troubleshooting	15
Index	19

Getting Started with DOE-Based Application Development

This developer guide provides information about using SAP® Mobile Platform features to create DOE-based applications.

SAP Data Orchestration Engine (DOE)

DOE is the short name for the SAP® Data Orchestration Engine, which is a component of NetWeaver Mobile. DOE consolidates data from SAP back-end sources and distributes the data to mobile applications.

DOE is event-driven middleware that receives data from the SAP back end, follows rules to calculate all the affected receivers, and prepares the data to be sent to the devices without waiting for the devices to connect. DOE uses messaging to transmit data to and from the devices.

DOE includes:

- An administration and monitoring component for managing and troubleshooting the data flow.
- A design-time component, the DOE Workbench, for defining data objects and data distribution.
- A runtime component that replicates, synchronizes, and distributes data between the system components.

DOE-based applications, working with SAP Mobile Platform, allow system designers to model and consolidate SAP mobile content in the middle tier, while separately layering distribution rules over this content. This approach is especially useful when back ends cannot provide a mobile interface that serves up mobile data, or if additional flexibility is required. Distribution rules can evolve separately from the content model, and different distribution rule sets can be used with the same content model.

When building a DOE-based application, the developer starts by using the DOE Workbench to describe back-end interaction and the application content model. After the content model is described and the distribution rules are configured, the developer can create a package to be deployed to SAP Mobile Server to allow devices to communicate with DOE.

Once the mobile package is deployed, the developer can generate device-side artifacts that form the basis of mobile application interactions with SAP Mobile Platform services and data. One or more packages can be used within a single application. Package version information is embedded in the device-side artifacts and is used to match the device application with the correct runtime package.

DOE-Based Application Scenarios

The SAP Mobile Platform DOE option consolidates all mobile data from back-end SAP systems, then uses rules to determine mobile distribution. This approach allows distribution rules to evolve separately from the content model and for different distribution rule sets to be used with the same content model.

The SAP Mobile Platform DOE deployment option allows the system designer to model and consolidate SAP mobile content in the middle tier and separately layer distribution rules over this content. This approach allows distribution rules to evolve separately from the content model and for different distribution rule sets to be used with the same content model. Even customers can change the rules without rewriting client or back-end code, after actively deploying applications.

The technology to enable this behavior is built directly into the NetWeaver stack and is therefore best suited to SAP-only implementations or where third-party back-end integration is already provided through NetWeaver. This method specifies BAPI CRUD interfaces to adapt back-end suite datasources to the middleware data consolidation area.

The SAP Mobile Platform DOE option consolidates all mobile data from the back-end SAP system, then uses rules to determine mobile distribution. The rules are fired on these events:

- New device is registered – initial receiver determination
- Back-end data or client data changes because of user updates or batch updates – staging are update
- Business change resulting in change of user subscriptions, for example, moving from region A to region B – device realignment

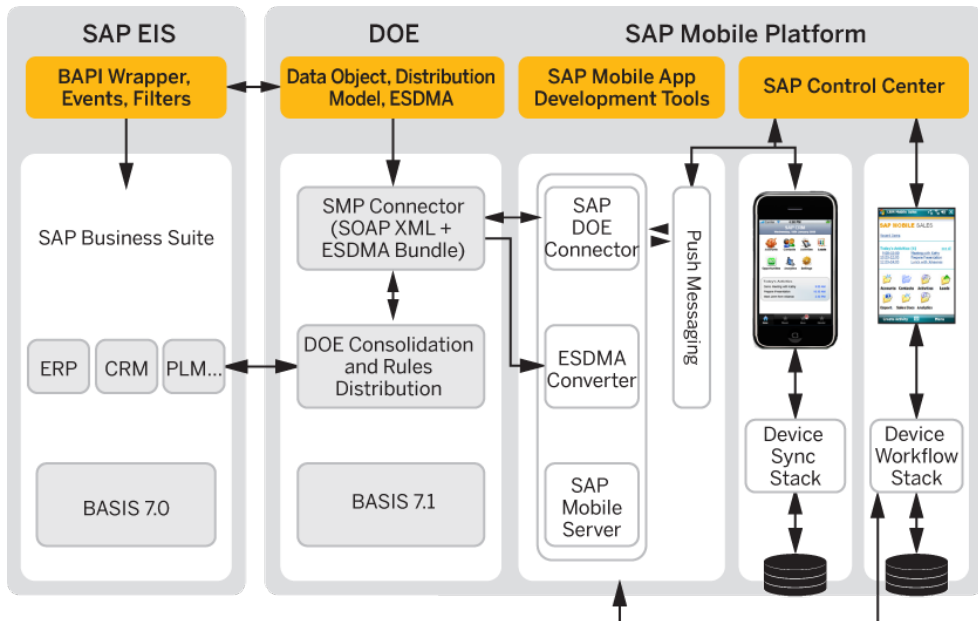
DOE-based applications are well suited to these situations:

- The SAP back end cannot directly support mobile queries, or mobile queries place an unacceptable load on the back end.
- The design dictates that the data distribution take place in the middleware.
- A scenario requires flexibility in adapting to different back-end interfaces or versions without changing the mobile model.
- A scenario requires flexibility in changing the distribution rules without changing the mobile model and mobile application.
- A scenario requires a centralized in-box for error resolution by an administrator.
- Multiple customized SAP back ends must work with the same mobile application, for example, if the same mobile application is resold to multiple customers who use different distribution rules.
- A mobile solution is working primarily against the SAP back end.
- Customized conflict resolution is required within the mobile middleware, rather than in the back end.

Application Architecture

All application data resides on the SAP back end. The Data Orchestration Engine (DOE) consolidates the data and communicates with SAP Mobile Platform, which communicates with the application on user devices. SAP Control Center manages the user devices.

Figure 1: System Architecture of a DOE-based Application



The SAP back end is where the enterprise data resides, in the SAP Business Suite, running on top of the SAP Basis infrastructure.

BAPI wrappers, events, and filters that you create in the DOE Workbench make the data available for distribution and consolidation by the DOE, through data objects and a distribution model that you also create in the DOE Workbench, and output to SAP Mobile Platform as an entity set definition for mobile applications (ESDMA) bundle.

On SAP Mobile Platform, the ESDMA converter utility converts the ESDMA bundle into an SAP Mobile Platform package that you deploy to SAP Mobile Servers. You use the SAP Mobile SDK and native integrated development environments to create the screens and business logic that ties the application together with a user interface, which you deploy to devices through the SAP Control Center.

On user devices, the application communicates through push messaging with the SAP® Data Orchestration Engine Connector (DOE-C), which relays data back and forth with the DOE.

The DOE consolidation and distribution rules ensure that only the absolute minimum of data necessary to keep the device data and the SAP back end data in sync is transmitted.

Monitoring of SAP Mobile Platform messages in the store and forward infrastructure takes place in SAP Control Center.

Developer Roles

There are three stages of DOE-based application development, each involving distinct types of tasks and skill sets. These correspond to three different developer roles.

In your organization, the same person may perform multiple roles.

SAP Expertise

Someone with an advanced knowledge of SAP systems must perform a considerable amount of work on the SAP back end to make the SAP data accessible to your mobile application.

The SAP back-end tasks are summarized in this document, and include links to more detailed instructions in the SAP documentation. See *Creating a Software Component Version* on page 8.

Administrative Expertise

The final task on the SAP back end is to generate an ESDMA bundle. Before you can begin device platform-specific development, this ESDMA bundle must be converted to a SAP Mobile Platform package, which is then deployed to one or more SAP Mobile Servers. See *Modeling Distribution Logic* on page 9.

Device Platform-Specific Expertise

The remaining tasks for creating a DOE-based application include generating native code that is specific for the target device platform, then bundling that code with supporting compile-time and runtime components, to build a functioning application. See *Developing Platform-Specific Device Code* on page 13.

SAP Data Structure Customization

You can perform varying degrees of customization on the data that comes from the SAP back end.

There are several levels of customization of the original SAP data structure.

Customization	Tasks to Implement	Tasks to Upgrade
None.	Download the ESDMA bundle and deploy it on SAP Mobile Server.	Subscribe a new device and start using the application.

Customization	Tasks to Implement	Tasks to Upgrade
Change the distribution pattern.	Change the distribution pattern, before or after downloading ESDMA bundle.	Unsubscribe and resubscribe old devices.
Add code-free extensions.	Add custom fields to dummy nodes on the SAP back end. Include code in the application to manage the dummy fields.	None; users continue using the application without interruption.
Change the data model.	Release the SWCV for shipment, then change the data object, enhance and regenerate the back-end adapter, and generate and download the ESDMA.	Generate a new application with a new ESDMA and deploy to SAP Mobile Server in a new domain. Continue running both old and new applications in different domains and let users decide when to move to the new application in the new domain.

Development Task Flow

Define the back-end interface in the SAP Business Suite, define data models and distribution logic in the DOE, develop mobile components in SAP Mobile Platform and in a platform-specific IDE, and deploy the application on SAP Mobile Platform.

	Defining Back-End Interface	Defining Data Model and Data Distribution Logic	Developing Mobile Components	Deploying the Application
SAP Business Suite	1. Identify or create BAPIs and encapsulate them			
DOE		2. Create software component versions 3. Model data objects 4. Model distribution logic	5. Create, generate, and download ESDMA	
SAP Mobile Platform			6. Convert ESDMA bundle to Unwired Platform package 7. Deploy package to SAP Mobile Server	11. Deploy finished application
Platform-specific IDE			8. Generate initial code 9. Add code for screens and business logic 10. Test and debug	

1. *Creating and Encapsulating BAPIs*

Use the BAPI Wrapper wizard to identify or write BAPI wrappers that make SAP data available to your application.

2. *Creating a Software Component Version*

The Software Component Version (SWCV) is a shipment unit for design-time objects in the Data Orchestration Engine (DOE) repository. You must create SWCVs before you can create data objects.

3. *Modeling Data Objects*

You must model data objects to make the required application data available to DOE for consolidation and distribution.

4. *Modeling Distribution Logic*

Perform distribution logic modeling tasks, which require system administrator expertise, in the back-end SAP system.

5. *Creating, Generating, and Downloading the ESDMA Bundle*

The ESDMA bundle contains SAP metadata in a form you can import into the SAP Mobile Platform environment on your developer workstation.

6. *Converting the ESDMA Bundle into an SAP Mobile Platform Package*

Use the ESDMA Converter utility to convert an ESDMA bundle into a SAP Mobile Platform Runtime package you can deploy to SAP Mobile Server.

7. *Deploying the SAP Mobile Platform Package*

Deploy the SAP Mobile Platform package to SAP Mobile Server before you begin coding your application. You will then be ready to test your code.

8. *Developing Platform-Specific Device Code*

To complete a DOE-based application, you must generate initial code, manually add code, test and debug the code, and deploy the finished application. These tasks are covered in platform-specific developer guides.

Creating and Encapsulating BAPIs

Use the BAPI Wrapper wizard to identify or write BAPI wrappers that make SAP data available to your application.

For detailed instructions, see the SAP documentation, *Creating BAPI Wrappers Using the Wizard*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/3e/1e6323e59a4dc1b236f4905960ccdd/content.htm

1. Identify the business scenario that you want to mobilize.
2. Define the flow of your application.
3. Use the BAPI Wrapper Wizard to identify or write BAPI wrappers.

Creating a Software Component Version

The Software Component Version (SWCV) is a shipment unit for design-time objects in the Data Orchestration Engine (DOE) repository. You must create SWCVs before you can create data objects.

For detailed instructions, see the SAP documentation:

- *Transport Organizer (BC-CTS-ORG)*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/57/38dd4e4eb711d182bf0000e829fbfe/frameset.htm
- *Creating a Software Component Version*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/e2/cba40f77a144658f8bb5181371a5d8/content.htm

1. Create a transport organizer.

2. Log in to the SAP NetWeaver Application Server and open the Data Orchestration Engine Workbench.
3. Choose **Create SWCV**.
4. Fill in the information required to create the SWCV.

Modeling Data Objects

You must model data objects to make the required application data available to DOE for consolidation and distribution.

For detailed instructions, see the SAP documentation:

- *Data Object Concepts*, at http://help.sap.com/saphelp_nwmobile711/helpdata/en/47/a991ef7dbd3376e10000000a421937/frameset.htm
- *Creating a Back-End Adapter*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/88/9e21142beb442b9eae5085e34e813c/frameset.htm
- *Customizing a Receiver Meta Model*, at http://help.sap.com/saphelp_nwmobile711/helpdata/de/48/74befbd1431b5ae10000000a42189c/frameset.htm

1. Create the data objects.
2. Create the back-end adapters.
3. Create the distribution patterns.
4. (Optional) Customize the receiver meta model.

Modeling Distribution Logic

Perform distribution logic modeling tasks, which require system administrator expertise, in the back-end SAP system.

For detailed instructions, see the SAP documentation:

- *Concepts of Distribution*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/6b/b2edf8d0b9463fabbea2e50d088182/frameset.htm
- *Creating Distribution Models*, at http://help.sap.com/saphelp_nwmobile71/helpdata/en/6b/b2edf8d0b9463fabbea2e50d088182/frameset.htm

1. Create a distribution model.
2. Create a distribution dependency.
3. (Optional) Add distribution rules to the distribution model.

Creating, Generating, and Downloading the ESDMA Bundle

The ESDMA bundle contains SAP metadata in a form you can import into the SAP Mobile Platform environment on your developer workstation.

For detailed instructions, see the SAP documentation, *Configuration Guide Gateway to SAP NetWeaver Mobile*, available from https://websmp202.sap-ag.de/~form/sapnet?_SHORTKEY=01100035870000729606&_SCENARIO=01100035870000000202&.

1. Create an ESDMA.
2. Assign data objects.
3. Assign a distribution model software component version (SWCV).
4. Define a design-time projection.
5. (Optional) Define runtime projections.
6. (Optional) Define a runtime header.
7. (Optional) Configure a client-specific distribution model SWCV (DMSWCV).
8. (Optional) Add custom services.

For example, specify the back-end search BAPI for an ESDMA object.

9. Generate the ESDMA bundle.
10. Download the .zip file for the ESDMA bundle to a temporary location on your developer workstation.
11. Extract the contents of the ESDMA bundle .zip file to an `<ESDMA_dir>` directory on your developer workstation.

Note: The `<ESDMA_dir>` directory name cannot contain any spaces.

Converting the ESDMA Bundle into an SAP Mobile Platform Package

Use the ESDMA Converter utility to convert an ESDMA bundle into a SAP Mobile Platform Runtime package you can deploy to SAP Mobile Server.

1. Create a META-INF directory under the `<ESDMA_dir>` where you downloaded and extracted the ESDMA bundle for your application.
2. In the META-INF directory, create a file named `sup-db.xml`.
3. Use a text editor to open `sup-db.xml`, and copy these lines into it:

```
<package name="EXAMPLE" short-name="EXAMPLE" sup-name="EXAMPLE"
version="1.0"
  java-package="com.sybase.example.db" /* For Android
```

```

Application */
  cs-namespace="Sybase.Example.db" /* For Windows Mobile
Application */
  oc-namespace="example_db_" /* For iOS Application */
  <!-- Update with new host and port, listener.url must end with /
doe/publish. -->
  <property name="listener.url" value="http://
SCC_HOME:<listener_port>/doe/publish" />
  <database name="example-database" />
  <database-class name="ExampleDatabase" />
  <personalization-parameter name="language" type="string"
owner="client" />

  <include file="afx-esdma.xml" />

</package>

```

4. Replace the bolded values above as follows:

- **EXAMPLE** – your SAP Mobile Platform package name. This value appears three times.
- **com.sybase.example.db** – the required Java namespace for BlackBerry application development.
- **Sybase.Example.db** – the required C# namespace for Windows and Windows Mobile application development.
- **example_db_** – the required Objective C namespace for iOS application development.
- **http://SCC_HOME:<listener_port>/doe/publish** – the SAP Mobile Server host name and listener port.

Note: The listener port must be followed by "/doe/publish", with no trailing "/".

The default value for the listener port is 8000.

- **example-database** – the name of the required database file.
- **ExampleDatabase** – the name of the required database class.

5. (Optional) Add additional entity definitions not defined in the ESDMA.

This is necessary to define client-side data entities that have no relationship with entities defined in the ESDMA bundle.

6. (Optional) In the `<ESDMA_dir>\Resources` directory, create a `MetaMerge.xml` file.

This file is used by the SAP Mobile Platform code generation and deployment tools to adapt the ESDMA bundle, if necessary to, for example:

- Hide select data objects defined in the ESDMA.
- Add supplementary indexes on data objects that can be used by the client.

7. Run the ESDMA converter utility to convert the ESDMA bundle resource metadata file to a SAP Mobile Platform package.

See *esdma-converter Command* in *System Administration*.

8. After running the ESDMA converter utility, verify that these files have been created in the META-INF directory:
 - `afx-esdma.xml`
 - `ds-doe.xml`

Deploying the SAP Mobile Platform Package

Deploy the SAP Mobile Platform package to SAP Mobile Server before you begin coding your application. You will then be ready to test your code.

You must use the **deploy** command in the DOE-C Command Line Utility to deploy a DOE-C package. This is the only DOE-C Command Line Utility command that is not available in the SAP Control Center.

In an SAP Mobile Platform cluster, deploy the package to the primary SAP Mobile Server node – SAP Mobile Platform automatically replicates the package to the other nodes.

To provide failover and load balancing in an SAP Mobile Platform cluster, specify the URL of a load balancer that is capable of routing to all the SAP Mobile Server nodes in the cluster.

You must specify the URL of a load balancer by setting the **listener.url** property in the META-INF\sup-db.xml file, where the ESDMA bundle was unzipped. See *Deploying the SAP Mobile Platform Package* in *Mobile Data Models: Using Data Orchestration Engine*.

In the steps below you are prompted for each of the parameters for the deploy command. Alternatively, you may include the parameters when entering the command. See *SAP DOE Connector Command Line Utility* in *System Administration*.

1. Start the SAP DOE Command Line Utility.

See *Starting the Command Line Utility Console* in *System Administration*.

2. At the **doec-admin** prompt, enter **deploy**, followed by the command parameters.

```
deploy -d|--domain domainName
-a|--applicationID appID
{-u|--technicalUser SAPUserAccount
-pw|--password SAPUserPassword} |
{-ca|--certAlias certificateAlias}
[-sc|--securityConfiguration securityConfigName]
[-dir|--deployFilesDirectory deploymentDirectory]
[-h|--help] [-sl|--silent]
```

Alternatively, enter **deploy**, then enter the parameters as prompted.

Developing Platform-Specific Device Code

To complete a DOE-based application, you must generate initial code, manually add code, test and debug the code, and deploy the finished application. These tasks are covered in platform-specific developer guides.

Continue with these tasks in the developer documentation for your specific platform:

- *Developer Guide: Android Object API Applications*
- *Developer Guide: iOS Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*

Documentation Roadmap for SAP Mobile Platform

SAP® Mobile Platform documents are available for administrative and mobile development user roles. Some administrative documents are also used in the development and test environment; some documents are used by all users.

See *Documentation Roadmap* in *Fundamentals* for document descriptions by user role.

Check the Product Documentation Web site regularly for updates: <http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories>, then navigate to the most current version.

Troubleshooting

Determine the causes of problems and apply the recommended solutions.

Problem	Resolution
Unable to parse error in DOE-C logs	<p>Possible causes:</p> <ol style="list-style-type: none">1. Duplicate SyncKey – verify that the failed XML uses unique SYNCKEY_MMW values across all nodes.2. Metadata mismatch – verify that:<ul style="list-style-type: none">• The DOE configuration for ESDMA is as expected.• If the ESDMA has been regenerated, the older ESDMA is undeployed from SAP Mobile Platform and the new one is deployed.• Ensure the code is regenerated, recompiled, and deployed to devices.

Problem	Resolution
<p>Packet dropped state with message, Request Entity Too Large</p> <p>Three consecutive entries in the log contain the text shown below – in the first entry, <code><actual_byteSize></code> is larger than <code><max_byteSize></code>.</p> <pre>... Detected a packet-dropped candidate message - size <actual_byteSize>, maximum <max_byteSize></pre> <pre>... DOE-C MESSAGE ID: 554 CODE: 413 MESSAGE: Request Entity Too Large ...is now in a packet-dropped state. The DOE-C will no longer push messages to this subscription.</pre>	<p>Use the setEndpointProperties command in the SAP® Data Orchestration Engine Connector Command Line Utility (CLU) to make the <code>doePacketDropSize</code> larger than the <code><actual_byteSize></code> in the first of the three consecutive error log entries.</p> <pre>setEndpointProperties {-i --in inputXmlFile} {-d --domain domainName {-a --all -ps --packageNames nameAndVersionList} {-u --technicalUser SAPUserAccount -pw --password SAPUserPassword} {-ca --certAlias certificateAlias} [-ds --doePacketDropSize byteSize] [-ew --doeExtractWindow maxNumMsgs] [-t --soapTimeout seconds] [-h --help] [-sl --silent]</pre> <p>Note: When you increase <code>doePacketDropSize</code> you should also increase the JVM heap size.</p> <p>For more details on setEndpointProperties, see <i>setEndpointProperties Command in System Administration</i>.</p>
<p>User cannot get application through the SAP portal</p>	<p>The SAP portal must be configured to allow users to get DOE-based applications without administrator assistance.</p> <ol style="list-style-type: none"> 1. Ask your SAP basis team to apply note 1250795 to the portal server, which enables the challenge pop-up. See https://websmp230.sap-ag.de/sap/support/notes/1250795. 2. Have users enter a URL in SAP Mobile Platform that is something like this: <code>http://<portalUrl:port>/redirect/redirect?url=/irj/portal</code>.

Problem	Resolution
Subscription times out	<p>Possible causes:</p> <ol style="list-style-type: none">1. During import, there are exceptions in processing the message. For example, Parsing Error.2. <code>importResult</code> is not being generated (Check SAP Mobile Server log with DOE-C debug enabled).3. The DOE server is sending all messages with <code>SER_MSG_ID = 0</code> (visible in SOAP dumps). This causes DOE-C to drop messages as duplicates. <p>Possible solutions:</p> <ul style="list-style-type: none">• For Parsing errors, identify the data causing the parsing failure (for example, causing a Duplicate Sync Key error) and re-subscribe.

Index

A

application architecture 3

B

back-end adapter 9

BAPIs 8

D

data model 9

developer roles 4

development task flow 7

distribution logic

 modeling 9

documentation roadmap 13

E

ESDMA bundle

 converting to to SAP Mobile Platform package
 10

 creating, generating, and downloading 10

G

getting started with DOE-based application

 development 1

M

modeling data 9

P

platform-specific device code 13

R

receiver meta model 9

S

SAP data structure customization 4

SAP Mobile Platform package

 converting from ESDMA bundle 10

 deploying to SAP Mobile Server 12

software component version (SWCV)

 creating 8

T

task flow

 development 7

troubleshooting 15

W

where DOE-based application is best 2

