Fundamentals

SAP Mobile Platform 2.3

DOCUMENT ID: DC01915-01-0230-02

LAST REVISED: April 2013

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

Contents

Solve Mobility Challenges with SAP Mobile Platform

Extending your enterprise business processes, data, and information away from the office is critical to success in today's business environment, however, achievement of a truly mobile enterprise entails many technical challenges. SAP® Mobile Platform provides a development and deployment platform that solves challenges, supports mobile applications to fit your enterprise mobility use case, and helps to balance device user requirements with enterprise requirements.

Mobility solutions present complexity to the IT organization. These complexities may include, but are not limited to, lack of integrated solutions, heavy upfront costs for large-scale customization requirements, and the lack of repeatable development and deployment models. SAP Mobile Platform addresses these complexities by providing a platform-based solution that allows you to:

- Leverage existing mobile development tools and expertise to develop mobile applications
- Develop mobile applications to meet your users' data needs and access data from a variety of enterprise information systems
- Deploy to a variety of mobile device platforms
- Configure and manage the runtime environment to complement areas of your existing infrastructure and security.

SAP Mobile Platform supports development and deployment of several categories of mobile applications to meet the enterprise system access needs of your business scenario.

Offline Mobile Applications

With offline mobile applications, information availability is mission critical regardless of connectivity. These applications remain operational even when they are not connected to the EIS. Data is synchronized between the Enterprise Information System (EIS) and the device when connectivity is available. Application requirements typically involve:

- The need to use data on the device if the application is offline
- More complex task orientation, data input, and screen navigation

Application developers have two options

- Object API development where developers write custom code — C#, Java, or Objective-C, depending on the target device platform — to create device applications and leverage platform specific services. These native applications leverage SAP Mobile Platform advanced runtime services for data synchronization management that typically requires data aggregation and distribution to diverse EIS.
- Agentry application development where developers build applications using a 4GL development interface for business logic, client behavior and functionality, and other

application behaviors, coupled with more traditional development languages that define the data synchronization with the EIS.

Hybrid Apps

Hybrid apps offer an option that is a hybrid of HTML5 and native development approaches. Hybrid apps provide a feature-rich web based user interface and the simplicity of container-based deployment. The typical use case for hybrid web container applications includes mobile workflows and lightweight applications. Human resource management applications for vacation or travel approval are examples of this occasionally connected application. Application requirements typically involve:

- Low data volume
- Simple, form-based user experience, involving notifications, lookup, and action-decision forms
- Light-weight persistence

Hybrid Container applications leverage SAP Mobile Platform services from within a standards based HTML(5)/JavaScript environment, developers create applications using Hybrid App Designer or using third party web tools. With this approach, a Hybrid Web Container is customized for each device type supported in a business mobility environment, and developers create HTML5 applications that perform advanced, device specific operations on all the different devices. The Hybrid Web Container embeds a browser control supplied by the device OS, which allows developers to build applications with simplicity of Web development but utilize the power of native device services.

Online Applications

With online mobile applications, information access is adhoc in nature and access to data is not available in areas without network access. Application requirements typically involve:

- Request/reply directly with the EIS
- Light-weight editing of form-based information
- Few screens and simple entries

Online applications are developed using the SAP® OData SDK, REST API, or Agentry Editor. These applications leverage SAP Mobile Platform primarily synchronous request/reply pattern for interaction with services. With OData SDK applications, developers create applications that allow users to work (create, update data), connecting the application to SAP® NetWeaver Gateway exposed data.

The types of application development and support with SAP Mobile Platform help to solve mobility challenges and meet mobility requirements in the enterprise.

See also

- *SAP Mobile SDK* on page 27
- *Object API Applications* on page 35
- *OData Applications* on page 40

- *HTML5/JS Hybrid Apps* on page 37

Enterprise Mobility Requirements

SAP Mobile Platform, with SAP Mobile SDK and SAP Mobile Platform Runtime, provides functionality and features to address enterprise requirements.

Connectivity

- Provides the middleware platform where users connect to and interact with various enterprise information systems and applications that exist in any large or medium size corporate IT infrastructure that supports customers, employees, sales, partners, suppliers, and executives. This includes SAP® back-end systems such as SAP Business Suite.

Security

- End-to-end security ensures that connections, applications, and data objects are deployed and managed securely.
- Data is partitioned for mobile applications; securing data while in transport and while accessed by the application.
- Data access and application functionality is available only to authorized mobile users.

Device and App Management

- Provides a common runtime infrastructure that supports deployment and management of a variety of mobile application types.
- Supports the rapidly expanding and changing selection of devices and device operating systems on the market.
- Implements a data model that fits the application requirements, physical restrictions and capabilities of the mobile device, such as network bandwidth, storage capacity, connectivity, security, and availability of consistent data.
- Reliably propagates user transactions initiated on a device that may or may not have network connectivity.
- Manages device and application life cycles, including device upgrades and turnover, deployment of application updates and new applications, with limited interruption to productivity.

Development Tools to Minimize Total Cost of Development

- Development tools leverage existing developer expertise, technologies, and commonalities, such as data models and objects, to effectively produce mobile applications for a variety of device types and device platforms.

Device User Requirements

To maximize device user productivity, mobile applications developed and deployed with SAP Mobile Platform support common device user requirements. Support of these user requirements are balanced with enterprise requirements.

Common device user requirements may include:

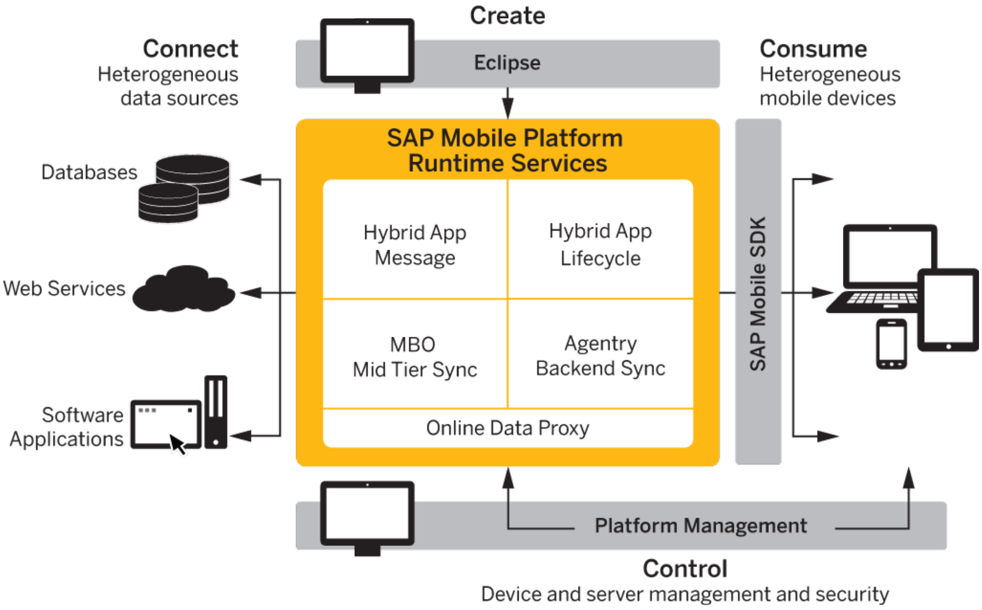
- Device choice. More organizations now support this as well as the ability to configure a personal device for work productivity and vice-versa.
- Ease of use. Devices and applications that are easy to configure and use. App interfaces must be intuitive and integrate with device features, such as automatic layout, touch screen or trackball navigation, etc.
- Flexibility. The ability to load and upgrade applications, receive notifications, and perform work when connectivity is unavailable.

Platform Overview

SAP Mobile Platform provides an integrated platform solution to extend your enterprise applications to mobile workers in the field.

SAP Mobile Platform acts as the hub that connects enterprise information systems and data sources to mobile devices. Features for mobile application development, deployment, security, and ongoing mobile device and mobile application management (MDM/MAM) provide a complete end-to-end solution. The platform is made up of:

- **SAP Mobile SDK –** The platform development tool set used to build mobile applications that meet your mobility needs.
    - SAP Mobile WorkSpace provides an Eclipse-based, integrated development for modeling, designing, and deploying mobile applications.
    - The Agentry Editor provides an Eclipse-based, integrated development environment for developing Agentry applications.
    - Libraries are also provided to support building Object API, HTML5/JS Hybrid, and OData SDK mobile applications.
- **SAP Mobile Platform Runtime –** The deployment and management architecture and services used to run and manage mobile applications. SAP Mobile Server is an engine that provides server-side runtime services, such as security, caching, and synchronization, to mobile applications and enables integration with the EIS.
    - SAP Control Center, a key component of Runtime, is the web-based console that provides platform management and control, which includes mobile device and mobile application monitoring and management..



This solution allows you to:

- **Connect –** During development and deployment, connect to your heterogeneous data sources and back-end enterprise systems.
- **Create –** Use the development tools included with the SAP Mobile SDK to build and test mobile applications that meet your mobility needs.
- **Control –** Deploy to and manage SAP Mobile Platform Runtime, including the runtime environment, end-to-end security, and device applications.
- **Consume –** Mobile applications install to devices allowing device users to work online and offline. Enterprise data is accessed from a variety of mobile devices.

The chapters in *Fundamentals* guide you through details of this mobility platform.

- *Documentation Roadmap*

  Use the documentation roadmap to review the documentation available for SAP® Mobile Platform and determine where to start based on your role.
- *Mobile Data Models*

  There are multiple ways to model enterprise data so that mobile applications can be developed to access data in your enterprise: Mobile Business Objects using the SAP Mobile SDK, Data Objects using SAP Data Orchestration Engine (DOE), OData using SAP® NetWeaver Gateway, and Object, Transaction, and Synchronization definitions using Agentry Editor.
- *Mobile Application Development*

SAP Mobile Platform supports multiple options for development and deployment of mobile applications.

- *SAP Mobile Platform Runtime*

  SAP Mobile Platform Runtime provides the platform infrastructure that allows you to deploy and manage your mobile applications.

Documentation Roadmap

Use the documentation roadmap to review the documentation available for SAP® Mobile Platform and determine where to start based on your role.

Bookmark *SAP Mobile Platform Product Documentation* at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&amp;c=firsttab&amp;a=0&amp;p=categories* to quickly access the complete set of online documentation and check for updates.

To complement the core product documentation, the following are available.

Tutorials, Projects and White Papers

Check SDN regularly for tutorials, projects and white papers: *http://scn.sap.com/docs/DOC-8803*.

Tutorials demonstrate how to use SAP Mobile Platform tools to mobilize your enterprise, using step-by-step instructions. They are available on Product Documentation, and SAP™ Development Network (SDN).

Projects are the completed output of each of the base tutorials that you can use to verify output from your tutorial. You can choose to use your own tutorial output or one of these projects as the base to start working on other tutorials.

Articles include white papers and supplementary technical documents about SAP Mobile Platform and related topics.

Samples

Check the SAP® Development Network (SDN) Web site regularly for new and updated samples: *https://cw.sdn.sap.com/cw/groups/sup-apps*.

Sample applications are fully developed, working applications that demonstrate the features and capabilities of SAP Mobile Platform. You can download and run these working applications.

Documentation by Role

The following descriptions assist in determining appropriate documents for review by the various roles within SAP Mobile Platform.

- *Documentation for Developers*

  If you are a developer, identify documentation based on your development focus.
- *Documentation for Administrators*

  Identify documentation based on administrative focus.

Documentation for Developers

If you are a developer, identify documentation based on your development focus.

Convenient diagrams are provided to guide you through documentation for the following development categories:

- *Mobile Data Models Development Documentation*

  Review documentation for developers who model data interaction from the EIS.
- *Native Object API Application Development Documentation*

  Review documentation for developers using the Object API to develop native applications for devices running Android, BlackBerry, iOS, and Windows and Windows Mobile operating systems.
- *HTML5/JS Hybrid App Development Documentation*

  Review documentation for developing Hybrid Web Containers and HTML(5)/JavaScript-based Hybrid Apps.
- *Native OData SDK Application Development Documentation*

  Review documentation for developers using the OData and Online Data Proxy APIs to develop native applications for devices running Android, BlackBerry, and iOS operating systems.
- *REST API Application Development Documentation*

  Review documentation for developers using a 3rd Party SDK with the Mobile Platform REST API.
- *Agentry Application Development Documentation*

  Review documentation provided for developers using the Agentry Editor to develop mobile applications.

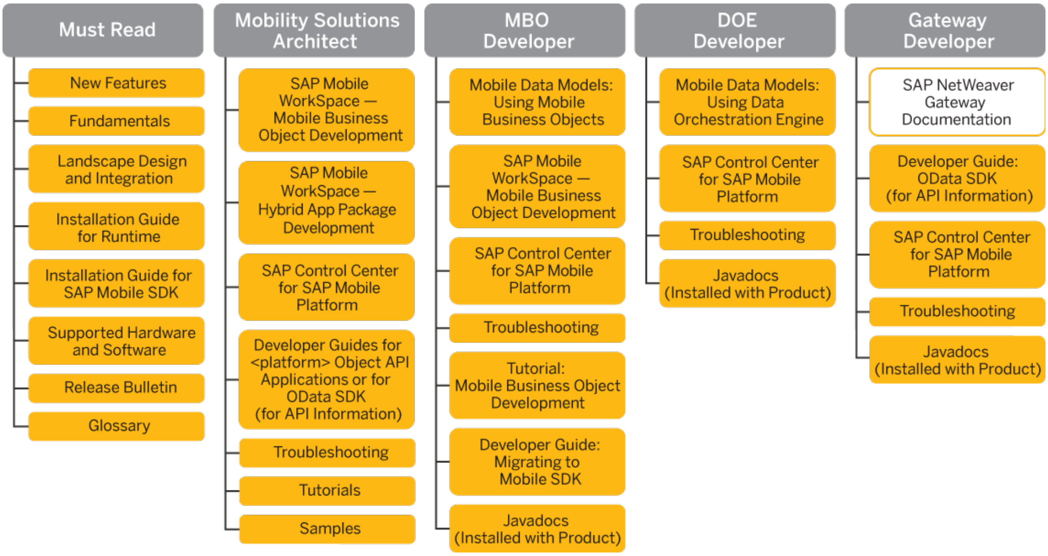
Mobile Data Models Development Documentation

Review documentation for developers who model data interaction from the EIS.

These roles encompass development of mobile data models:

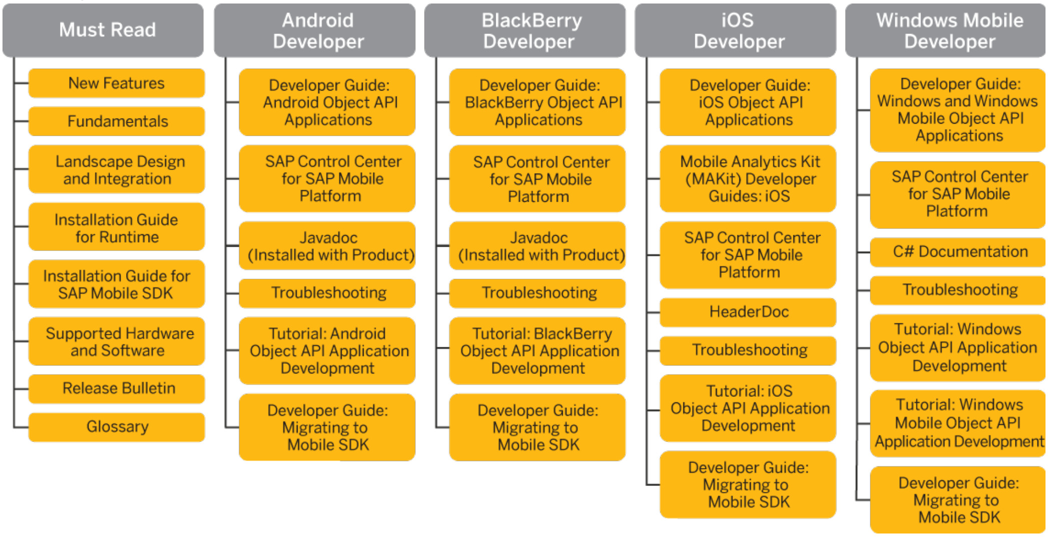
- **Mobility Solutions Architect –** Uses the tools to model the interaction between the mobile applications and existing EIS, services, and applications, or integrate mobile applications developed with other tools. Understands mobility concepts.
- **Mobile Business Objects (MBO) Developer –** Develops mobile business objects using SAP Mobile WorkSpace. Generates native application code for Android, BlackBerry, Windows, Windows Mobile, and iOS.
- **Data Orchestration Engine (DOE) Developer –** Uses the DOE workbench to develop data objects.

- **SAP NetWeaver Gateway and OData Developer –** Creates SAP NetWeaver Gateway content using the Gateway Content Generator, identifies existing content exposed by Gateway, or writes custom code. See SAP NetWeaver Gateway documentation at: *http://help.sap.com/nwgateway*


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
Mobility Solutions Architect
SAP Mobile WorkSpace – Mobile Business Object Development
SAP Mobile WorkSpace – Hybrid App Package Development
SAP Control Center for SAP Mobile Platform
Developer Guides for <platform> Object API Applications or for OData SDK (for API Information)
Troubleshooting
Tutorials
Samples
MBO Developer
Mobile Data Models: Using Mobile Business Objects
SAP Mobile WorkSpace – Mobile Business Object Development
SAP Control Center for SAP Mobile Platform
Troubleshooting
Tutorial: Mobile Business Object Development
Developer Guide: Migrating to Mobile SDK
Javadocs (Installed with Product)
DOE Developer
Mobile Data Models: Using Data Orchestration Engine
SAP Control Center for SAP Mobile Platform
Troubleshooting
Javadocs (Installed with Product)
Gateway Developer
SAP NetWeaver Gateway Documentation
Developer Guide: OData SDK (for API Information)
SAP Control Center for SAP Mobile Platform
Troubleshooting
Javadocs (Installed with Product)

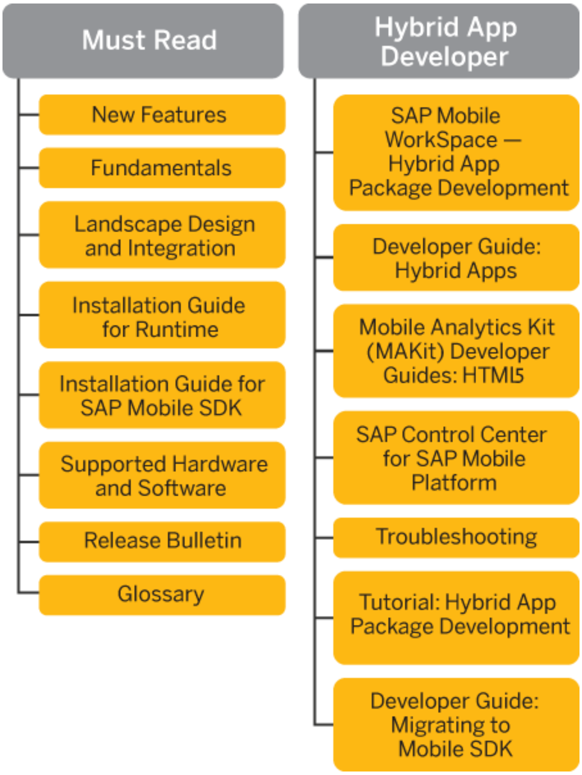
Native Object API Application Development Documentation

Review documentation for developers using the Object API to develop native applications for devices running Android, BlackBerry, iOS, and Windows and Windows Mobile operating systems.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
Android Developer
Developer Guide: Android Object API Applications
SAP Control Center for SAP Mobile Platform
Javadoc (Installed with Product)
Troubleshooting
Tutorial: Android Object API Application Development
Developer Guide: Migrating to Mobile SDK
BlackBerry Developer
Developer Guide: BlackBerry Object API Applications
SAP Control Center for SAP Mobile Platform
Javadoc (Installed with Product)
Troubleshooting
Tutorial: BlackBerry Object API Application Development
Developer Guide: Migrating to Mobile SDK
iOS Developer
Developer Guide: iOS Object API Applications
Mobile Analytics Kit (MAKit) Developer Guides: iOS
SAP Control Center for SAP Mobile Platform
HeaderDoc
Troubleshooting
Tutorial: iOS Object API Application Development
Developer Guide: Migrating to Mobile SDK
Windows Mobile Developer
Developer Guide: Windows and Windows Mobile Object API Applications
SAP Control Center for SAP Mobile Platform
C# Documentation
Troubleshooting
Tutorial: Windows Object API Application Development
Tutorial: Windows Mobile Object API Application Development
Developer Guide: Migrating to Mobile SDK

HTML5/JS Hybrid App Development Documentation

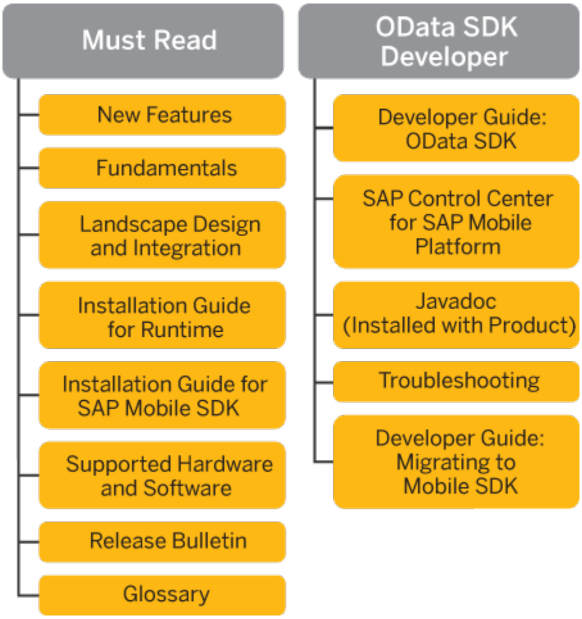
Review documentation for developing Hybrid Web Containers and HTML(5)/JavaScript-based Hybrid Apps.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
Hybrid App Developer
SAP Mobile WorkSpace — Hybrid App Package Development
Developer Guide: Hybrid Apps
Mobile Analytics Kit (MAKit) Developer Guides: HTML5
SAP Control Center for SAP Mobile Platform
Troubleshooting
Tutorial: Hybrid App Package Development
Developer Guide: Migrating to Mobile SDK

Native OData SDK Application Development Documentation

Review documentation for developers using the OData and Online Data Proxy APIs to develop native applications for devices running Android, BlackBerry, and iOS operating systems.

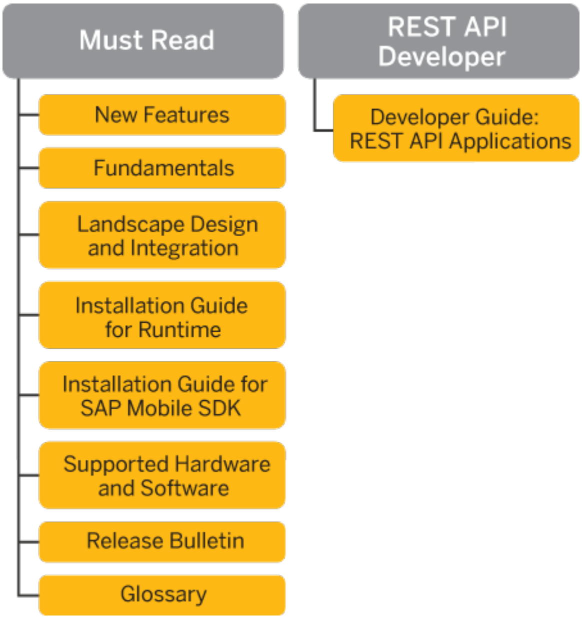
Also reference SAP Mobile NetWeaver Gateway documentation at: *http://help.sap.com/nwgateway*.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
OData SDK Developer
Developer Guide: OData SDK
SAP Control Center for SAP Mobile Platform
Javadoc (Installed with Product)
Troubleshooting
Developer Guide: Migrating to Mobile SDK

REST API Application Development Documentation

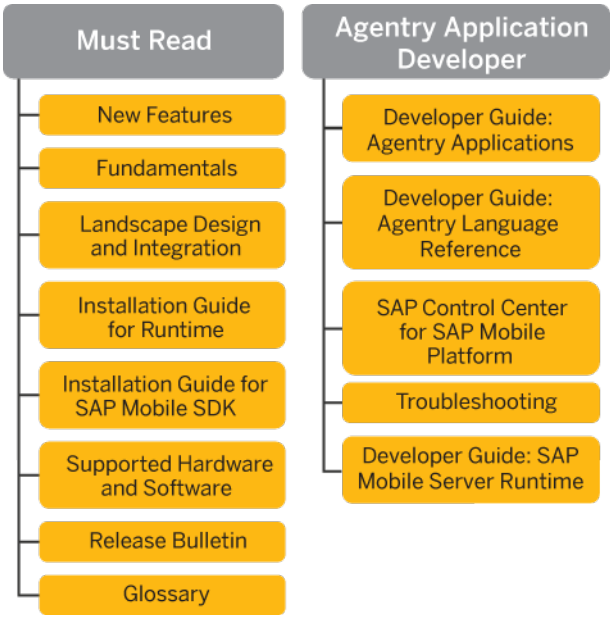
Review documentation for developers using a 3rd Party SDK with the Mobile Platform REST API.

Also reference *Mobile Data Models Developer documentation* and documentation available for SAP NetWeaver Gateway documentation at: *http://help.sap.com/nwgateway*


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
REST API Developer
Developer Guide: REST API Applications

Agentry Application Development Documentation

Review documentation provided for developers using the Agentry Editor to develop mobile applications.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
Agentry Application Developer
Developer Guide: Agentry Applications
Developer Guide: Agentry Language Reference
SAP Control Center for SAP Mobile Platform
Troubleshooting
Developer Guide: SAP Mobile Server Runtime

Documentation for Administrators

Identify documentation based on administrative focus.

Runtime administration encompasses multiple roles:

- **System Administrator; Domain Administrator; Security Administrator –** These responsibilities may be carried out by one or more individuals, depending on the size and complexity of the enterprise:
  - Plans deployment of production system, and monitors and maintains system health. Understands system configuration.
  - Plans, creates, configures, and administers one or more hosted domain. Each domain may have multiple tenants.
  - Performs ongoing monitoring and maintenance of the cache database (CDB) and other runtime databases.
  - Plans and implements enterprise level security – access, roles, policies.
- **Mobile Application Administrator –** Responsible for operation of the mobile applications running inside the system, and in the field. Understands mobility concepts.

Convenient diagrams are provided to guide you through documentation for the following administration categories:

- *Runtime Administration Documentation*
- *External Systems Administration Documentation*
- *Administration Automation Development Documentation*

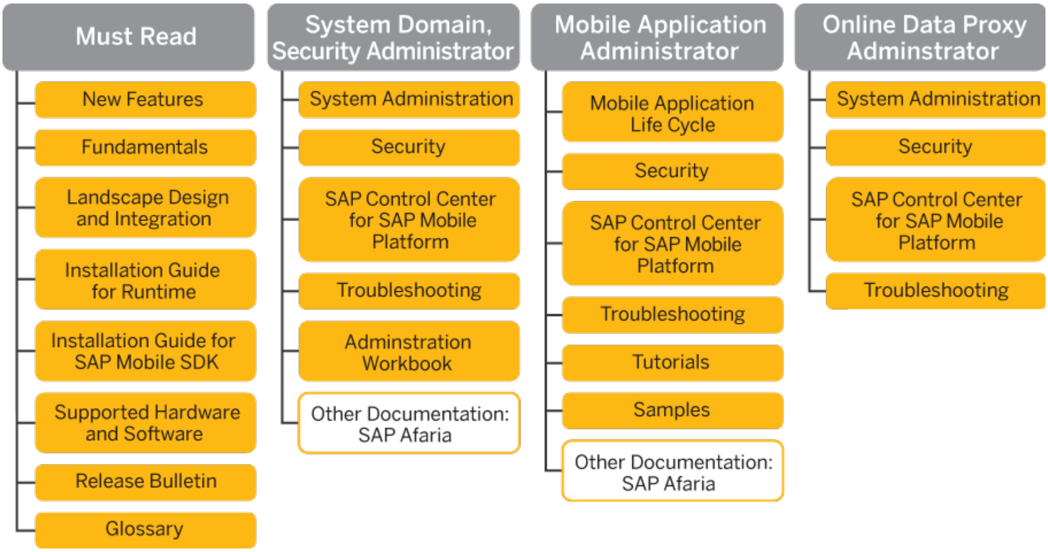
Runtime Administration Documentation

Runtime administration encompasses multiple roles:

- **System Administrator; Domain Administrator; Security Administrator –** These responsibilities may be carried out by one or more individuals, depending on the size and complexity of the enterprise:
  - Plans deployment of production system, and monitors and maintains system health. Understands system configuration.
  - Plans, creates, configures, and administers one or more hosted domain. Each domain may have multiple tenants.
  - Performs ongoing monitoring and maintenance of the cache database (CDB) and other runtime databases.
  - Plans and implements enterprise level security – access, roles, policies.

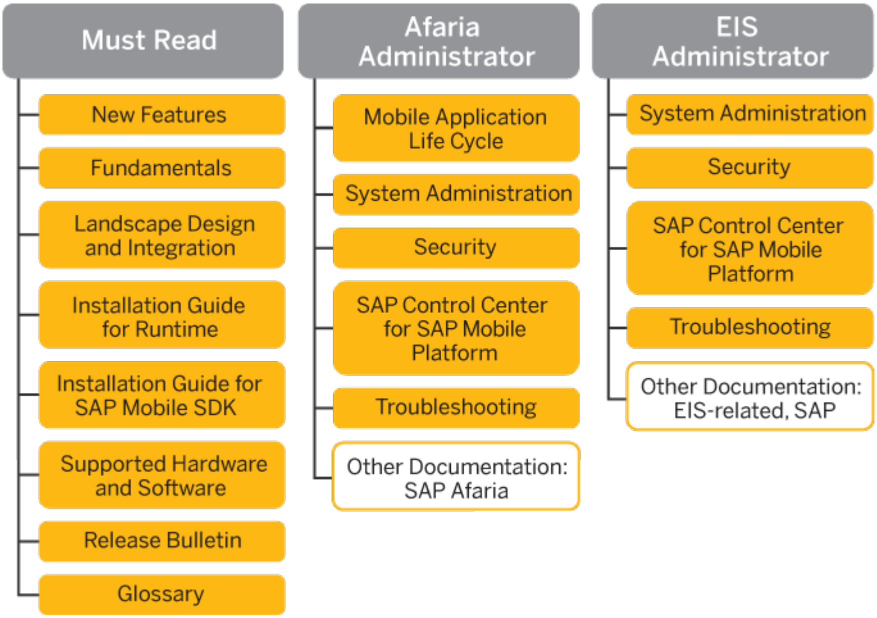
- **Mobile Application Administrator –** Responsible for operation of the mobile applications running inside the system, and in the field. Understands mobility concepts.

Documentation for Afaria is available at: *https://frontline.sybase.com/support* . Your Afaria-customer Mobile Enterprise Technical Support logon is required.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
System Domain, Security Administrator
System Administration
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting
Adminstration Workbook
Other Documentation: SAP Afaria
Mobile Application Administrator
Mobile Application Life Cycle
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting
Tutorials
Samples
Other Documentation: SAP Afaria
Online Data Proxy Administrator
System Administration
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting

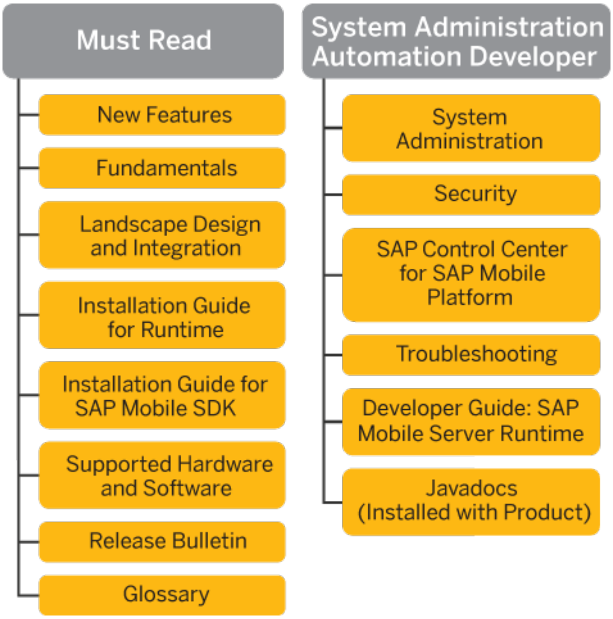
External Systems Administration Documentation

Administrators for systems external to SAP Mobile Platform need to understand aspects of how their systems specialty integrates with SAP Mobile Platform.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
Afaria Administrator
Mobile Application Life Cycle
System Administration
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting
Other Documentation: SAP Afaria
EIS Administrator
System Administration
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting
Other Documentation: EIS-related, SAP

Administration Automation Development Documentation

Within the context of administration, the administration automation developer fills a development role.


Must Read
New Features
Fundamentals
Landscape Design and Integration
Installation Guide for Runtime
Installation Guide for SAP Mobile SDK
Supported Hardware and Software
Release Bulletin
Glossary
System Administration Automation Developer
System Administration
Security
SAP Control Center for SAP Mobile Platform
Troubleshooting
Developer Guide: SAP Mobile Server Runtime
Javadocs (Installed with Product)

Mobile Data Models

There are multiple ways to model enterprise data so that mobile applications can be developed to access data in your enterprise: Mobile Business Objects using the SAP Mobile SDK, Data Objects using SAP Data Orchestration Engine (DOE), OData using SAP® NetWeaver Gateway, and Object, Transaction, and Synchronization definitions using Agentry Editor.

Based on the mobile application category that meets your user and data needs, and prior to defining your data models, identify the mobile application archetype you will use for development.

Category	Mobile Application Archetype	Data Model
Offline applications	Native Object API applications Agentry Applications	Mobile Business Object or Data Orchestration Engine Agentry
Hybrid Apps	HTML5 / JS Hybrid apps	Mobile Business Object or OData
Online applications	OData SDK applications or REST API applications Agentry Applications	OData or other business data model Agentry

Based on your data model and business logic, the EIS interface must be defined. This process eventually involves adapting existing services exposing data from the EIS. Your data models connect with your EIS using these data services.

- *Mobile Business Objects*

  Mobile Business Objects (MBOs) are developed to define the mobile data model to be used by Object API Applications and HTML5/JS Hybrid Apps. Developers build the data model using the Mobile WorkSpace graphical tools, which are provided with SAP Mobile SDK. These tools simplify and abstract EIS connections, and provide a uniform view of transactional objects. MBOs are reusable, allowing you to leverage them across multiple mobile device types.
- *OData*

  The SAP NetWeaver Gateway exposes OData with extensions specific to SAP. Service documents, that describe service interaction and data, allow users to interact with the SAP application.
- *Data Orchestration Engine*

  When the back-end enterprise system includes the Data Orchestration Engine (DOE) component of SAP NetWeaver Mobile, DOE data objects define the mobile data model for

Object API Applications. DOE consolidates data from diverse SAP back-end sources and distributes it to applications via SAP Mobile Platform messaging.

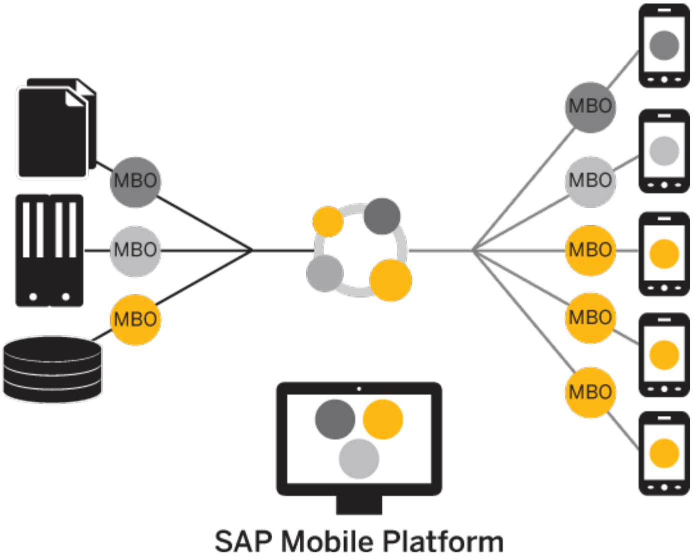
- *Agentry*

  Agentry applications handle data modeling and development in a single task. The developer creates Object, Transaction, and Synchronization definitions within the application project using the Agentry Editor’s 4GL development interface.

Mobile Business Objects

Mobile Business Objects (MBOs) are developed to define the mobile data model to be used by Object API Applications and HTML5/JS Hybrid Apps. Developers build the data model using the Mobile WorkSpace graphical tools, which are provided with SAP Mobile SDK. These tools simplify and abstract EIS connections, and provide a uniform view of transactional objects. MBOs are reusable, allowing you to leverage them across multiple mobile device types.

When deployed using SAP Mobile Platform, MBOs provide a layer of abstraction from SAP Mobile Server's interaction with heterogenous EIS/devices, as shown in the following diagram.



The following sections describe general concepts about Mobile Business Objects. For additional information about designing and developing MBOs, see:

- *Mobile Data Models: Using Mobile Business Objects*
- *SAP Mobile WorkSpace - Mobile Business Object Development*
- *Tutorial: Mobile Business Object Development*

See also

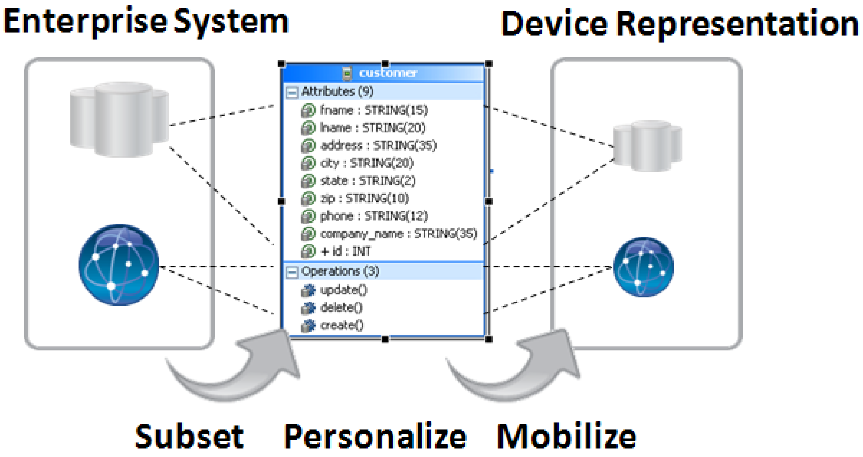
- *Object API Applications* on page 35
- *HTML5/JS Hybrid Apps* on page 37

Data Modeling

After connecting to datasources using connection profiles from SAP Mobile WorkSpace, the MBO developer defines the mobile interaction pattern, which typically involves selecting data subsets.

The MBO is developed to define each data subset, describe the data and the operations on the data.

- The data subset is deployed with the MBO package to SAP Mobile Server where the MBO manages synchronization between the EIS and SAP Mobile Server.
- Artifacts generated from the MBO are then used to develop the mobile application, typically further defining the data subset for data representation on the device.



Operations

MBOs may incorporate operations that change the data retrieved from the enterprise information system (EIS).

- Create, Update, Delete (CUD) operations – an operation definition contains arguments that map to the arguments of the EIS operation, and can create, update, or delete data. These operations cause state change.

  Operation replay executes the client-initiated transactions (Create, Update, and Delete operations) and propagates those changes to the EIS.

- Read/Load – an operation that includes optional load arguments that determine initial loading of data from the EIS to the SAP Mobile Server cache. For example, a SQL Select statement for database datasources.
- Entity Read – an operation's cache policy determines how the results of operation replay are applied to the SAP Mobile Server cache. An Entity Read operation can be defined to refresh a single instance of an MBO, including an MBO object graph (all MBOs linked by composite relationships).
- OTHER operation – an operation definition for operations other than create, update, or delete. These operations do not cause state change.

The operation definition supports validation and error handling.

Other Key Concepts

Other key concepts for understanding mobility include object queries, synchronization parameters, result set filters, result checkers, and personalization keys.

- Object queries – a SQL statement associated with a mobile business object (MBO) that runs on the client against data that was previously downloaded to the device. The object query searches the device database and returns a filtered result set (such as a single row of a table). Object queries enable discrete data handling.
- Synchronization parameters – metadata that defines how the values provided by the device application client are used to filter data already existing in the CDB, which is downloaded to the device, to provide data of interest to the user. Synchronization parameters may be used to retrieve the cached data, or mapped to load arguments to filter the data that is cached for the MBO and then served to the client application.
- Result-set filters – a Java API used to customize the data returned by an enterprise information system (EIS) operation. Developers can use result-set filters to alter or manipulate returned data and its structure, that is, MBO attributes, before it is placed in the server's cache.
- Result checkers – a Java API that implements operation execution checks, and error handling logic for the MBO operation. Developers can use result set checkers to implement customized error checking for MBOs, since not all MBOs use a standard error reporting technique.
- Personalization keys – metadata that enables users to store their name/value pairs on the device persistent store, the client application session (in memory), or the SAP Mobile Server, and use the personalization keys to provide values to load arguments, synchronization parameters, operation arguments, or device application business logic specific usages. To use a personalization key for filtering, it must be mapped to a synchronization parameter/load argument.

  The developer can define personalization keys for the application, or use two built-in system defined personalization keys—username and password—to perform single sign-on from the device application to the SAP Mobile Server, authentication and authorization on SAP Mobile Server, as well as connecting to the back-end EIS using the same set of credentials. The password is never saved on the server.

Data Caching

Data caching is initial loading, or filling, the SAP Mobile Server cache with enterprise information system (EIS) data, then continuing to refresh the cache with changes from the EIS or mobile device on an ongoing basis.

Since continual synchronization of the data between the EIS and device puts a load on the EIS, SAP Mobile Platform provides several options for loading and refreshing the data cache.

Options include narrowing the EIS data search so that only specific data is retrieved (based on load arguments), identifying effective policies for handling data updates once operations are performed (based on an operation's cache policy), scheduling periodic updates to occur when system usage is low (based on cache group and refresh policy), updating only changed data in the cache, and so forth.

You can use multiple options to load and refresh the right data at the right time, and to deliver the smallest, most focused payload to the mobile device.

The primary loading schemes provide a trade-off between time and storage space. For example, bulk loading takes more time because data is loaded for all users, but once loaded, the data can be shared between users.

- Bulk loading – data for all users is loaded in bulk.
- Client defined partitioned data loading – only user relevant data, or partition, is loaded, based on MBO load arguments.
- EIS defined partitioned data loading – only user relevant data, or partition, is loaded, based on the DCN `partition` attribute.
- On-demand loading – data is not filled, until the client performs synchronization.
- Scheduled loading – data is filled periodically according to a schedule you set.
- Event notifications – data is pushed by the EIS when specific events occur.

Data change notification (DCN) facilitates propagation of data changes from the back-end enterprise information system (EIS) to the SAP Mobile Server interface for any MBO. The DCN payload containing changed data is applied to the cache and the change gets published to subscribing device users based on a change detection interval time. DCN is used for cases where the cache refresh may be expensive due to volume. This option is the least intrusive and most optimal for addressing high-load environments and optimizing the load on the EIS to keep the SAP Mobile Server cache consistent.

Object API Code Generation

To access and integrate MBOs in a device application, developers generate object code for the target device platform, and then use their IDE of choice to build the native device application. The object code generation step is the bridge from the SAP Mobile Server server-side development (MBOs) to client-side development (device applications).

The generated object code for each MBO follows a standard pattern of properties for attributes, operations, and abstracts persistence and synchronization. Object code generation is supported in the native language for each target platform. SAP Mobile Server client libraries complement and are required for the generated object code, which together are used in the device application.

The generated code is built upon the Mobile client libraries, which combine support for reliable transmission of data and transactions, security of data while in transit or on device, sending notifications when data changes occur in the back-end application, consistent interface on all platforms, all of which abstract developers from mobility related complexities.

Code generation is supported for these platforms: BlackBerry and Android (Java), iOS (Objective-C), Windows Mobile and Windows (C#).

See *Supported Hardware and Software* for the most current version information.

Package Deployment

The last step of mobile business object (MBO) development is to deploy the MBO definitions to SAP Mobile Server as a deployment package.

When you deploy MBOs to the SAP Mobile Server, you are deploying:

- MBO definitions including attributes, operations, connections, synchronization groups, and cache configuration as defined in the package.
- MBO custom code related to result-set filters and result checkers.
- Other functionality captured in the MBO model.

MBOs are deployed using a deployment wizard through which you can make the choices that are appropriate for application requirements. Developers use SAP Mobile WorkSpace to deploy a package.

The production administrator can deploy from a wizard using the web-based management console, or from the command line. Deployment-time tasks include choosing:

- Target domain – logical container for packages.
- Security configuration – used for authentication and authorization of users accessing the package.
- Server connections mapping – to bind MBOs design-time datasources to production datasources.
- Application ID – the application ID registered for the application. The value differs according to application client type.

OData

The SAP NetWeaver Gateway exposes OData with extensions specific to SAP. Service documents, that describe service interaction and data, allow users to interact with the SAP application.

Where an OData source is used as the back-end the application developer does not need to create any model elements (MBOs) in the tooling but rather inherits a service model from the service document published from SAP NetWeaver Gateway. These OData service documents contain all the information the device developer needs to parse and interact with these data streams.

See documentation for SAP NetWeaver and for SAP NetWeaver Gateway for additional information.

See also

- *OData Applications* on page 40

Data Orchestration Engine

When the back-end enterprise system includes the Data Orchestration Engine (DOE) component of SAP NetWeaver Mobile, DOE data objects define the mobile data model for Object API Applications. DOE consolidates data from diverse SAP back-end sources and distributes it to applications via SAP Mobile Platform messaging.

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.

For additional information, see documentation for:

- *Mobile Data Models: Using Data Orchestration Engine*
- SAP NetWeaver Mobile, including DOE

See also

- *Object API Applications* on page 35

Agentry

Agentry applications handle data modeling and development in a single task. The developer creates Object, Transaction, and Synchronization definitions within the application project using the Agentry Editor’s 4GL development interface.

Synchronization logic, referenced by the synchronization definitions, is developed in the language which corresponds to the enterprise information system the mobile application extends. This can include Java interfaces or API’s; database systems using SQL; web services via web (CGI) requests that return structured XML data parsed by the Agentry Server using XPaths; and the host system upon which the SAP Mobile Platform in general, and Agentry Server specifically, is installed.

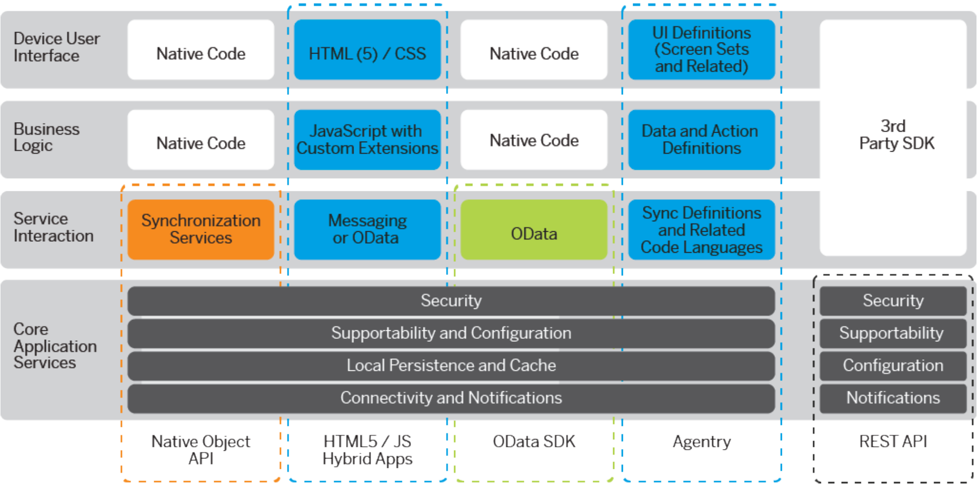
Mobile Application Development

SAP Mobile Platform supports multiple options for development and deployment of mobile applications.

Two primary development paths, each with one or more application archetypes, or development modes, are supported:

- Development using SAP Mobile SDK. Four different archetypes are supported: Object API Applications, HTML(5)/JS Apps, OData SDK Applications, and Agentry Applications. All provide access to core application services, application specialization, and HTML(5)/JS within the development SDK. Native code development may be needed for business logic and the device user interface.
- Development using a 3rd party SDK, such as Sencha or PhoneGap, with appropriate modifications to use the Mobile Platform REST API. With this option, all development takes place outside of SAP Mobile SDK but uses the REST APIs to access core application Web services exposed via SAP Mobile Server.

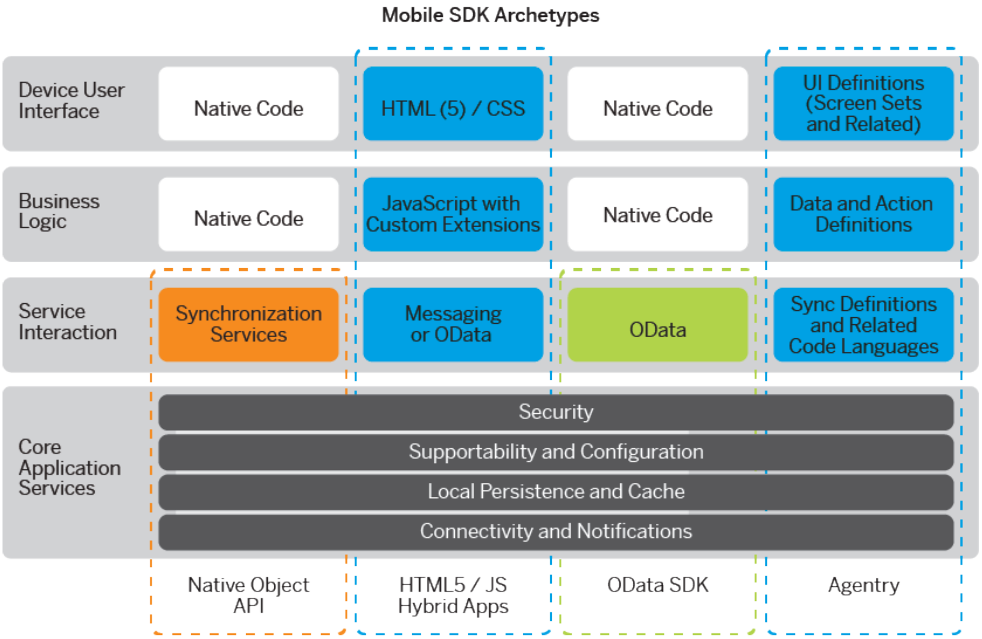
Mobile Application Archetypes



SAP Mobile SDK

SAP Mobile SDK supports the development lifecycle for Object API Applications, HTML5/JS Hybrid Apps, OData SDK Applications, and Agentry Applications.

The end result of the development process is a packaged mobile application ready for provisioning to devices. The following diagram shows the device application structure and services provided for each of the development approaches with SAP Mobile SDK.


Mobile SDK Archetypes
Device User Interface
Native Code
HTML (5) / CSS
Native Code
UI Definitions (Screen Sets and Related)
Business Logic
Native Code
JavaScript with Custom Extensions
Native Code
Data and Action Definitions
Service Interaction
Synchronization Services
Messaging or OData
OData
Sync Definitions and Related Code Languages
Core Application Services
Security
Supportability and Configuration
Local Persistence and Cache
Connectivity and Notifications
Native Object API
HTML5 / JS Hybrid Apps
OData SDK
Agentry

Dashed outlines in the diagram represent the services provided by SAP Mobile SDK for each application archetype. Native code development takes place in a third party development tool, such as Visual Studio, Eclipse, etc., outside of the SAP Mobile SDK. Native code development uses SAP Mobile Platform APIs that support core services and application specialization, in addition to leveraging in-house device platform development expertise.

- **Core Application Services –** The set of common services provides a consistent foundation upon which mobile apps are developed. These common device services make development easier and more efficient in developing multiple applications.

  Security features are embedded into the SDK to support secure storage of certificates, use of the artifacts in authentication, single sign-on, and other features related to encryption of the database. APIs are supplied to support certificate store, logon certificate, and the data vault. Each application type makes use of the same set of security features.

  Supportabilty and configuration allow developers to enable and configure exception reporting within the mobile application so that device users and developers can identify and troubleshoot issues.

  Local persistence and cache management define how the application stores and receives/ retrieves data and interacts with the middleware.

  Connectivity and notifications facilitate online/offline connections and push notifications to device users.
- **Service Interaction –** Service interaction is the differentiator between development approaches. Based on these distinct services and your business needs, develop one or more application types.

Develop Apps with	When your business needs
Native Object API SDK	Offline applications: Mission-critical and more complex. Users work on data offline and synchronize updates to the server when connected. Synchronization implies an intermediary store of cached data and the process by which data changes are propagated between the server, the cache, and the device storage with the goal of ensuring that the data is replicated in each location.
HTML5/JS Hybrid App SDK	Online applications with Push: Simple queries and workflows that follow a request-reply or lookup pattern. The HTML5/JS Container with Messaging or OData protocol are supported.
OData SDK	Online applications with Push that use OData protocol: Data, modeled in SAP NetWeaver Gateway, is accessed in a request-reply or lookup pattern.
Agentry Applications	Offline applications, where more complex business logic and synchronization uses the Exchange Data model. Online applications where real-time data can be accessed and Agentry Push is supported.

Push leverages a proprietary message transport to move data between device and server. The data change is initiated, or pushed, from wherever the data change takes place: from the device if the change takes place there, or from the server if the change takes place there.

- **Business Logic and Device User Interface –** The user interface is developed:
  - Using the developer IDE of choice for the target device platform and either the Object API or the OData SDK API.
  - Using the Hybrid App Designer to design HTML5/JS.
  - Using the Agentry Editor to develop the metadata UI, scripts and rules, and backend sync or request/reply..

The following sections provide detail about the development lifecycle and about each type of archetype development.

- *Development Life Cycle*

  The "development life cycle" takes developers through the process of designing, developing, testing, deploying and maintaining mobile business objects, device applications, and workflow packages.
- *Object API Applications*

  Object API applications are customized, full-featured mobile applications that use mobile data model packages, either using mobile business objects (MBOs) or Data Orchestration Engine, to facilitate connection with a variety of enterprise systems and leverage synchronization to support offline capabilities.

- *HTML5/JS Hybrid Apps*

  HTML5/JS Hybrid Apps support simple business processes, such as approvals and requests, and also use mobile business objects (MBOs) to facilitate connection with a variety of enterprise systems. With this approach, a hybrid web container is developed and deployed to a device, then one or more workflows are deployed to the container. This approach supports mobile workflow enablement, which enables mobile device users to operate as workflow participants, allowing the mobile user to start and respond to back-end enterprise requests within a generic framework.
- *OData Applications*

  OData applications are simple, mostly online mobile business applications that leverage standards-based approaches—OData protocol and RESTful Web Services. This approach provides instant quick value applications meant for end user productivity.
- *Agentry Applications*

  Mobile applications built using the Agentry Editor can be deployed directly to SAP Mobile Platform. Agentry applications already in production can be modified to run on SAP Mobile Platform.

See also

- *Solve Mobility Challenges with SAP Mobile Platform* on page 1

Development Life Cycle

The "development life cycle" takes developers through the process of designing, developing, testing, deploying and maintaining mobile business objects, device applications, and workflow packages.

SAP Mobile SDK is intended to work with your existing development paradigm. Use SAP Mobile SDK as you would use any development tool—integrated with your existing change control and versioning mechanisms, and with your development, test, and production environments.

Life Cycle Stages

Life cycle refers to the distinct stages of application development and upgrade. In the context of this document, life cycle refers to the application life cycle changes, focusing especially on initial release of the application, additional releases, and patches to the production system.

This is not a comprehensive description of the application development life cycle, but focuses on important information for making decisions about whether to create a new release, or simply patch the production system.

Figure 1: Life Cycle Development for Milestone Development

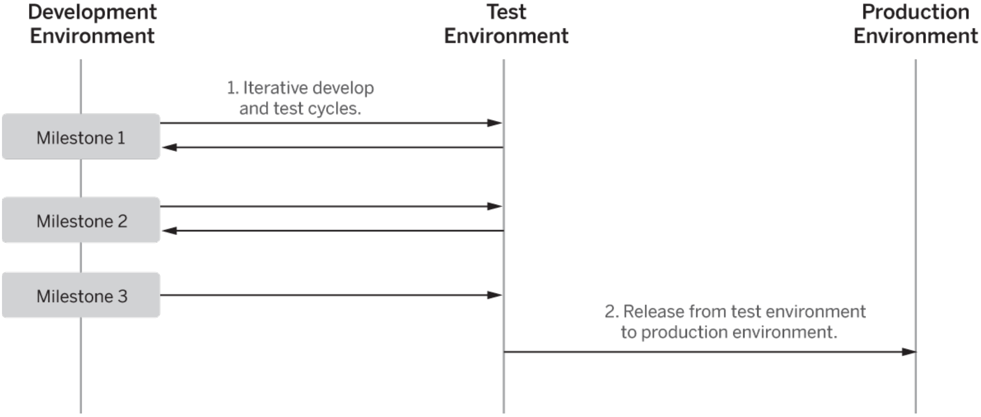
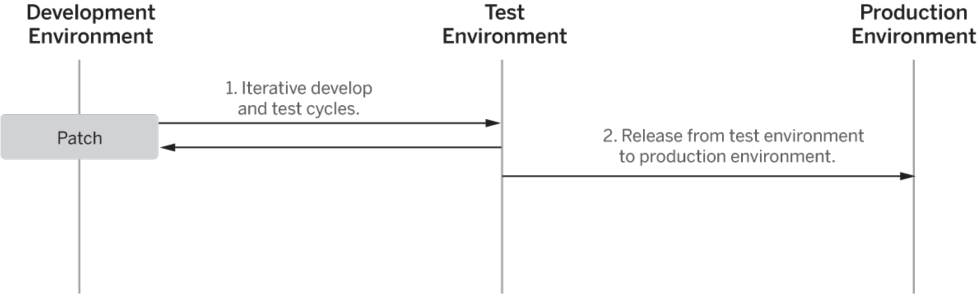

Development Environment
Test Environment
Production Environment
1. Iterative develop and test cycles.
Milestone 1
Milestone 2
Milestone 3
2. Release from test environment to production environment.

Figure 2: Life Cycle Development for Patches


Development Environment
Test Environment
Production Environment
1. Iterative develop and test cycles.
Patch
2. Release from test environment to production environment.

Stage 1: Development

During development, developers use tools to perform requirements analysis, design and model a solution, and develop application code.

Developers change code on an ongoing basis to develop their projects over time. This development process may be iterative, with defined milestones. For each milestone, developers work from stable milestone base on a local workstation and make additions or changes. Typically code is checked in and out of change control, and may be merged into a main code line periodically.

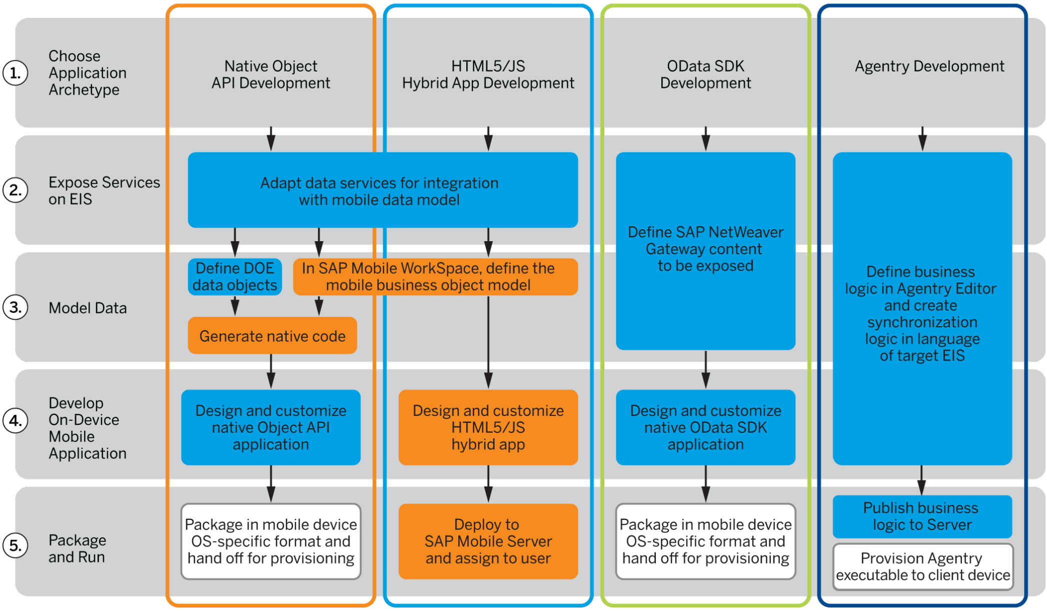
During the development stage:

- The model is typically deployed into a dedicated single or shared development SAP Mobile Server for testing.
- For an ongoing project, the base for a development cycle is typically the base from a previous cycle.

- The result of a development cycle is an intermediate release—or deployment package. An intermediate release could be a release or patch for a production system, or the base for more development.

Five steps characterize the development process. Details of each step vary based on the application archetype to be developed.

Development Task Flow


1. Choose Application Archetype
2. Expose Services on EIS
3. Model Data
4. Develop On-Device Mobile Application
5. Package and Run
Native Object API Development
HTML5/JS Hybrid App Development
OData SDK Development
Agentry Development
Adapt data services for integration with mobile data model
Define DOE data objects
In SAP Mobile WorkSpace, define the mobile business object model
Generate native code
Define SAP NetWeaver Gateway content to be exposed
Define business logic in Agentry Editor and create synchronization logic in language of target EIS
Design and customize native Object API application
Design and customize HTML5/JS hybrid app
Design and customize native OData SDK application
Package in mobile device OS-specific format and hand off for provisioning
Deploy to SAP Mobile Server and assign to user
Package in mobile device OS-specific format and hand off for provisioning
Publish business logic to Server
Provision Agentry executable to client device

Stage 2: Testing

During the development process, code may be tested on an iterative basis—typically at specific milestones—by quality assurance engineers.

The test environment should emulate the production environment, to ensure quality throughout the product life cycle. Any defects are reported back to developers, to be fixed. This iterative development and testing cycle continues as long as necessary to produce a high-quality product.

As part of the testing phase, you may need to upgrade the test environment itself as the project develops. During the transition, you may need to consider upgrade scenarios as part of the testing process.

Characteristics of the testing stage:

- Installing into and upgrading the test landscape is equivalent to installing into and upgrading the production landscape.
- The base for a testing cycle is typically an earlier release.

- The result of a testing cycle is a release candidate, or another iteration of development and testing.

Testing in the Development Environment

Test each native application and workflow package on a simulator or emulator, and make adjustments. Then, test in a test environment that emulates the production environment for complete end-to-end testing.

Download emulators for devices such as Windows Mobile, and use them from the SAP Mobile WorkSpace development environment. Alternatively, you can use Microsoft synchronization software, such as Windows Mobile Device Center, to connect the device to your development computer, and deploy and test the application on the device.

Download simulators for devices such as BlackBerry and iOS:

- For BlackBerry, device simulators and the BlackBerry Email and MDS Services Simulator Package are available.
- For iOS, the iOS simulator is provided as part of the Xcode environment.

After testing is complete on simulators or emulators, work with administrators to complete end-to-end testing in an environment that emulates the production environment.

Stage 3: Production

The production system changes on an ongoing basis. After testing, the release can be delivered to the production environment for use.

Typically this is a phased approach, where the system administrator installs a new version, or installs the latest version, and then migrates users individually to the new version. A provisioning tool, such as Afaria®, may be used to help with the roll out to users.

Characteristics of the production stage:

- During production, code changes are not allowed.
- Changes to the production system require a new development cycle (in some cases, patches, or hot fixes, can be applied to the production system).
- The base for a production cycle is typically an earlier release.

Stage 4: Maintenance

Changes to the production system require a new development effort. In some cases, a patch, or hot fix, may be applied to the production system.

Characteristics of the maintenance stage:

- During production, or even during testing, a fix for a defect is issued.
- The base for a patch is an intermediate release, or a release.

Best Practices

Implement best practices for developing, testing, and deploying device applications using the application life cycle.

Implement Three Environments

To produce robust, efficient, and high-quality mobile applications, implement SAP Mobile Platform using three environments—development, test, and production.

The test environment should replicate your production environment as closely as possible. Use the same components, such as an administrative console, security and authentication cluster configuration, and eventual use of Afaria, relay server, and clustering; network communications and protocols; and device simulators and actual devices.

It is important that you test mobile applications on devices and device simulators and emulators to ensure quality, but it is equally important to test the entire data handling experience between the device, SAP Mobile Server, and the enterprise information system (EIS) datasource.

During the test cycle, test engineers should work closely with developers to correct defects. This is typically an iterative process resulting in a production-quality product.

SAP Mobile Platform provide specific supportability features to follow the data path through the system. Use an administrative console, log files, and error messages to help validate your testing.

SAP highly recommends that you use automated tests for repeatability. To help isolate problems, you may want to code test scripts.

See also

- *Object API Application Development* on page 36
- *HTML5/JS Hybrid App API Development* on page 38
- *OData Application Development* on page 41
- *Supportability* on page 51

Implement Change Control

Implement change control as part of your development process. Integrate your change control system with SAP Mobile Platform. Ensure that all your development artifacts are routinely checked into your change control system, so you can revert back to a known version if necessary.

Eclipse includes a built-in client for the Concurrent Versions System (CVS). See the Eclipse FAQ to learn more:*http://wiki.eclipse.org/index.php/CVS_FAQ*. You can also integrate other source control systems, such as Subversion.

When implementing change control:

- Set up the repository for the shared development environment.
- Identify the artifacts to be maintained as source code, for example:
    - Model files (the Mobile WorkSpace project file)
    - Native application code, and generated and custom client code
    - Mobile WorkSpace artifacts, which are the MBO projects, custom result checkers, enterprise information system (EIS), and model files
    - Custom ResultSet filters and error checkers
    - EIS connection profiles
    - Projects
- Identify and communicate change control policies and procedures to the development team.

Object API Applications

Object API applications are customized, full-featured mobile applications that use mobile data model packages, either using mobile business objects (MBOs) or Data Orchestration Engine, to facilitate connection with a variety of enterprise systems and leverage synchronization to support offline capabilities.

The Object API application model enables developers to write custom code — C#, Java, or Objective-C, depending on the target device platform — to create device applications.

Development of Object API applications provides the most flexibility in terms of leveraging platform specific services, but each application must be provisioned individually after being compiled, even for minor changes or updates.

Development involves both server-side and client-side components. SAP Mobile Server brokers data synchronization and transaction processing between the server and the client components.

- Server-side components address the interaction between the enterprise information system (EIS) data source and the data cache. EIS data subsets and business logic are encapsulated in artifacts, called mobile business object or Data Orchestration Engine packages, that are deployed to the SAP Mobile Server.
- Client-side components are built into the mobile application and address the interaction between the data cache and the mobile device data store. This can include synchronizing data with the server, offline data access capabilities, and data change notification.

These applications:

- Allow users to connect to data from a variety of EIS systems, including SAP® systems.
- Build in more complex data handling and logic.
- Leverage data synchronization to optimize and balance device response time and need for real-time data.
- Ensure secure and reliable transport of data.

See also

- *Data Orchestration Engine* on page 25
- *Mobile Business Objects* on page 20
- *Solve Mobility Challenges with SAP Mobile Platform* on page 1

Object API Application Development

The Object APIs form the core building block of Object API Applications and provide the common set of APIs for consistency across device platforms, thus following the "design once and deploy anywhere" paradigm of building applications.

Object API categories for Object API Application development include:

- Object APIs that support access to MBO and DOE-based mobile data models. The APIs offer abstraction for accessing data from the EIS, sending operation executions to the EIS, updating and deleting data, and accessing persistent store in the object-oriented paradigm, among other things.
- Object APIs that offer consistent access to core application services across all device platforms. The APIs provide abstraction for the complex technical details of reliable communication between the device and SAP Mobile Server. This includes a variety of network conditions for optimal transmission of data and state information, connection APIs to the back-end infrastructure, secure access to the SAP Mobile Server data, publishing of data notifications and processing of such notifications, data protection, and access to UI management APIs and custom UI controls, such as signature control, among other things.

The common development task flow for Object API Applications includes these steps:

1. Verify that mobile data models have been developed and are available.
2. Configure the platform-appropriate development environment, such as Android SDK, BlackBerry JDE, Xcode for iOS, and Visual Studio for Windows and Windows Mobile.
3. Generate Object API code for mobile business objects and import the artifacts—libraries and code—to the development environment.
4. Develop the device application.
5. Package the application code and prepare the package for deployment.
6. Test the device application using a simulator/emulator.
7. Deliver the device application package to the SAP Mobile Platform Administrator for production testing and provisioning.

Complete details of this development approach and Object API are documented in *Developer Guide: <Device Platform> Object API Applications*.

See also

- *Building Mobile Analytics into Object API Applications* on page 37
- *Implement Three Environments* on page 34
- *HTML5/JS Hybrid App API Development* on page 38

- *OData Application Development* on page 41
- *Supportability* on page 51

Building Mobile Analytics into Object API Applications

The Mobile Analytics Kit (MAKit) component of SAP Mobile SDK provides graphical controls with analytics capabilities for mobile devices. Developers can use this component to develop mobile applications that offer sophisticated and compelling interfaces that let users easily view enterprise data.

The MAKit component is available for iOS Object API application development.

MAKit includes a built-in, optimized, on-device analytics engine, which avoids frequent, time-consuming queries to remote servers. The data visualization suite of charts currently includes 5 types - column, bar, pie, line and bubble. In addition to the five charts, MAKit also supports a table control.

See *Mobile Analytics Kit (MAKit) Developer Guides: iOS*

See also

- *Object API Application Development* on page 36

HTML5/JS Hybrid Apps

HTML5/JS Hybrid Apps support simple business processes, such as approvals and requests, and also use mobile business objects (MBOs) to facilitate connection with a variety of enterprise systems. With this approach, a hybrid web container is developed and deployed to a device, then one or more workflows are deployed to the container. This approach supports mobile workflow enablement, which enables mobile device users to operate as workflow participants, allowing the mobile user to start and respond to back-end enterprise requests within a generic framework.

These applications:

- Manage a low data volume
- Provide a simple user experience
- Implement simple business logic
- Avoid the need for long-lasting, offline, stateful transactions
- Ensure secure and reliable transport of data

With Hybrid App development, the server-side of the application is metadata-driven and the client-side of the application is a fully generated Web application package. The focus is on how data is rendered to the device user; data is made available using a request-response pattern, without synchronization. Lightweight applications and mobile workflows are developed to provide business logic and interaction with MBOs.

A native container application is packaged with a Web Browser plug-in and built-in core application services such as connectivity, guaranteed messaging, caching, and security. The container is provisioned to mobile devices. Data transport and access relies on messaging protocol between the server and the container on the device, invoking either online operations to the back end, or cached MBO data on the Unwired Server.

The mobile workflow package is compiled —consisting of platform-independent HTML, JavaScript and CSS resources— and can be deployed automatically to the container, without writing any code. Device and application services include offline cache, reliable messaging, and secure store.

The container is deployed to a device and provides the runtime from which request-response decision patterns are executed. It supports three basic patterns where one, or a combination of these patterns is applied to implement each use case:

- **Notifications –** A server-initiated action, performed in the EIS in the context of a business process, sends a notification to device users
- **Lookup –** Device users initiate a request for information from the EIS
- **Action/Decision Forms –** Device users submit a form to make a decision, such as an approval, that results in some EIS business process state transition

The container hosts an embedded browser and launches the individual mobile workflow applications. The workflows are assigned to users by administrators. Once assigned, those workflows can be initiated by the user (client-initiated) or automatically triggered as a result of a back-end event that is sent to the SAP Mobile Server as a data change notification request (server-initiated).

See also

- *Mobile Business Objects* on page 20
- *Solve Mobility Challenges with SAP Mobile Platform* on page 1

HTML5/JS Hybrid App API Development

An HTML5/JS App includes both the Mobile Workflow— business logic, the data itself, and associated metadata that defines data flow and availability, and the Container — device-resident presentation and logic. Within SAP Mobile Platform, development tools enable both aspects of HTML5/JS Hybrid App development.

The Container is customized natively to define and manage device presentation and logic. Integration with PhoneGap allows you to link your own custom native code to the Hybrid Web Container and call this native code from JavaScript, as well as access native device functionality using the PhoneGap framework.

Hybrid apps can reference:

- One or more MBOs and can include load parameters, personalization, and error handling
- OData sources using the Datajs library
- OData sources and MBOs together in a Hybrid App

.

Once you have developed MBOs and deployed them to SAP Mobile Server, or your OData sources, develop device-resident presentation and logic for your Hybrid App using the Hybrid App Forms Editor.

The common development task flow for HTML5/JS Hybrid Apps includes these steps:

1. Verify that mobile business objects have been developed and are available.
2. Start the developer environment: launch SAP Mobile WorkSpace, import the project containing MBOs, connect to SAP Mobile Server and deploy the project.
3. Open the Hybrid App Forms Editor and design your worklfow.
4. Package the hybrid app code and prepare the package for deployment.
5. Test the hybrid app using a simulator/emulator.
6. Deliver the hybrid app package to the SAP Mobile Platform Administrator for production testing and provisioning.

Complete details of this development approach and HTML/JS Hybrid App API are documented in:

- *SAP Mobile WorkSpace - Hybrid App Package Development*
- *Developer Guide: Hybrid Apps*

.

See also

- *Building Mobile Analytics into HTML5/JS Hybrid Apps* on page 39
- *Implement Three Environments* on page 34
- *Object API Application Development* on page 36
- *OData Application Development* on page 41
- *Supportability* on page 51

Building Mobile Analytics into HTML5/JS Hybrid Apps

The Mobile Analytics Kit (MAKit) component of SAP Mobile SDK provides graphical controls with analytics capabilities for mobile devices. Developers can use this component to develop mobile applications that offer sophisticated and compelling interfaces that let users easily view enterprise data.

The MAKit component is available for HTML5/JS hybrid apps.

MAKit includes a built-in, optimized, on-device analytics engine, which avoids frequent, time-consuming queries to remote servers. The data visualization suite of charts currently includes 5 types - column, bar, pie, line and bubble. In addition to the five charts, MAKit also supports a table control.

See *Mobile Analytics Kit (MAKit) Developer Guides: HTML5*

See also

- *HTML5/JS Hybrid App API Development* on page 38

OData Applications

OData applications are simple, mostly online mobile business applications that leverage standards-based approaches—OData protocol and RESTful Web Services. This approach provides instant quick value applications meant for end user productivity.

OData applications are developed to leverage a proxy connection to SAP EIS systems, such as SAP Business Suite, and are deployed using the Online Data Proxy runtime option, essentially making realtime SAP data available to device users. The proxy connection uses the RESTful OData protocol to connect through SAP Mobile Server and SAP NetWeaver Gateway to SAP enterprise information systems. Data is made available using a request-response pattern, without synchronization.

The proxy connection uses OData with SAP additions:

- OData is the Microsoft-owned open protocol —a resource-based web protocol for querying and updating data. It defines operations on resources using HTTP verbs (GET, PUT, POST, and DELETE) and identifies those resources using a standard URI syntax. Data is transferred over HTTP using the Atom or JSON format.
- OData for SAP Products provides SAP Extensions to the OData protocol that enable users to build user interfaces for accessing the data published via OData. This allows interfaces to define human-readable, language-dependent labels for all properties and free-text search within and across collections of similar entities (OpenSearch).

OData applications running on mobile devices use semantic annotations to tell the client which of the OData properties contain specific data, such as a phone number, a part of a name or address, or something related to a calendar event. This allows seamless integration with contacts, calendar, and telephony on the mobile device. A metadata document defines whether an entity set is searchable, which properties may be used in filter expressions, and which properties of an entity will always be managed by the server.

These applications:

- Connect mobile devices more directly to SAP business systems, essentially making OData applications online. Data is not persisted, but can be cached.
- Use SAP data and business process models
- Leverage subscription and push notification features
- Implement SAP single sign-on (SSO)
- Ensure secure and reliable transport of data.

See also

- *Solve Mobility Challenges with SAP Mobile Platform* on page 1
- *OData* on page 24

OData Application Development

The OData application development approach uses an OData source as the back end and allows device users to query and update data in an essentially online application.

The OData application developer inherits a service model from the service document published from SAP NetWeaver Gateway. These OData service documents contain all the information the developer needs to parse and interact with these data streams.

Developers use APIs related to components for parsing, caching, persistence, connectivity, and configuration to develop OData Applications.

The common development task flow for OData application development includes these steps:

1. Verify that service documents have been published from SAP NetWeaver Gateway and are available.
2. Configure the platform-appropriate development environment, such as Android SDK, BlackBerry JDE, and Xcode for iOS.
3. Develop the device application.
4. Package the application code and prepare the package for deployment.
5. Test the device application using a simulator/emulator.
6. Deliver the device application package to the SAP Mobile Platform Administrator for production testing and provisioning.

Complete details of this development approach and OData API are documented in *Developer Guide: OData SDK*.

See also

- *Implement Three Environments* on page 34
- *Object API Application Development* on page 36
- *HTML5/JS Hybrid App API Development* on page 38
- *Supportability* on page 51

Agentry Applications

Mobile applications built using the Agentry Editor can be deployed directly to SAP Mobile Platform. Agentry applications already in production can be modified to run on SAP Mobile Platform.

Agentry applications provide the ability to support one or more client device types and one or more enterprise information systems of varying types from a single application project, high levels of configuration and modification with minimal user impact, simple or complex multi-screen workflows, and high data volumes stored on client devices. These applications are built using a 4GL development interface for business logic, client behavior and functionality, and other application behaviors, coupled with more traditional development languages that define the data synchronization with the EIS.

Agentry applications can operate in environments that include only occasional network connectivity, or take advantage of real-time synchronization in environments with reliable, always-available network connectivity. A wide range of application types are supported by this model, including work management, inventory management, CRM operations, as well as many others.

Application requirements typically involve:

- Complex business logic for client behavior and functionality and/or data synchronization
- High volume data storage on the client device
- Multiple device platform support for a single application
- Potential for on-going modification of the mobile application based on changing implementation environment requirements over the application lifecycle

Agentry applications are built and maintained using the Agentry development environment, the primary tool for which is the Agentry Editor. The Agentry Editor runs as a plug-in to a standalone Eclipse environment, separate from SAP Mobile WorkSpace, and leverages this robust development environment's support for related development tasks involving, potentially, Java, SQL, or other languages. With this approach, pre-built Agentry Client executables are deployed to the client devices in the runtime environment, which then consume the business logic from the Agentry Server running in the SAP Mobile Platform environment, which then dictates all of the client-side behavior and functionality.

For additional information, see *Developer Guide: Agentry Applications*.

Agentry Application Development

The Agentry application development approach uses the Agentry Editor, an Eclipse plug-in installed in an Eclipse instance apart from SAP Mobile WorkSpace, as the primary development tool. This tool presents a 4GL development interface. All aspects of the mobile application behavior, including client UI, business rules enforcement, and data synchronization with EIS, are controlled using this tool by creating "definitions," which are the primary development component to an Agentry application project.

Data synchronization with the EIS is further defined using the language matching the EIS back end type, such as a Java API or database system using SQL. This logic is referenced by definitions within the application project which control when, how, and why this logic is processed and executed.

As a means of exposing the data within the mobile application to this logic, the developer makes use of the Syclo Data Markup Language (SDML) for all EIS types. Additionally, for Java logic, the Agentry Java API provides objects and methods which expose this same data.

The common development task flow for Agentry Applications includes these steps:

1. Analyze your EIS system and determine the data encapsulation needed for the mobile application

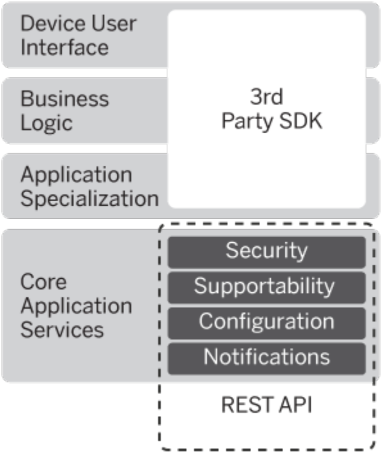
2. Analyze the needed data to design the proper synchronization logic and needs of the mobile application
3. Determine the proper data flow and, from there, screen flow on the client device for end users.
4. Create the definitions within the application project to represent the data, synchronization steps (though not necessarily the actual logic just yet), and user interface.
5. Develop the synchronization logic for the corresponding definitions within the application project.
6. Publish the application from the Agentry Editor to the Agentry Server. (This is the equivalent to a compile-build-link phase using a traditional language).
7. Deploy the Agentry Client executable to the client devices. Note that this deployment or device provisioning does not include the business logic developed in the Agentry Editor, and that the Agentry Client is a pre-built application executable provided with the SAP Mobile Platform.
8. Perform the initial transmit from the Agentry Clients to the Agentry Server, which results in the synchronization of the business logic for the mobile application as well as the initial retrieval of the production data from the EIS. This can either be performed as part of the implementation, or by the mobile users.

REST API Applications

Applications developed using 3rd Party SDKs, such as Sencha and PhoneGap, for any device platform, and using any programming language that supports sending and receiving HTTP messages can use the SAP Mobile Platform REST APIs to call services provided by the SAP Mobile Platform Runtime.

This development approach supports:

- Registration (creating an application connection)
- Authentication
- Native push notification
- Configuration



You build your client applications using third party developer tools, and without using any SAP Mobile SDK or native client libraries. SAP Mobile Platform provides management and monitoring of the applications, and support for native push notification such as Apple Push Notification Service, BlackBerry Enterprise Server push, or Google Cloud Messaging.

Starting 2.2 SP03, the REST SDK libraries have been introduced in the iOS and Android platforms to enable consumption of SAP Mobile Platform REST services with pure HTTPS connectivity. The REST SDK provides simplified APIs for registration, exchange settings between client and server, and end-to-end tracing. The SDK also supports native push notifications.

Optionally, you may use certain SAP Mobile Platform native client libraries to enable complex application capabilities, such as OData parsing and building, caching, or data vault management. You may also use third party JavaScript helper libraries such as those from Sencha, Appcelerator, or Adobe PhoneGap. For more information, see *Developer Guide: OData SDK*.

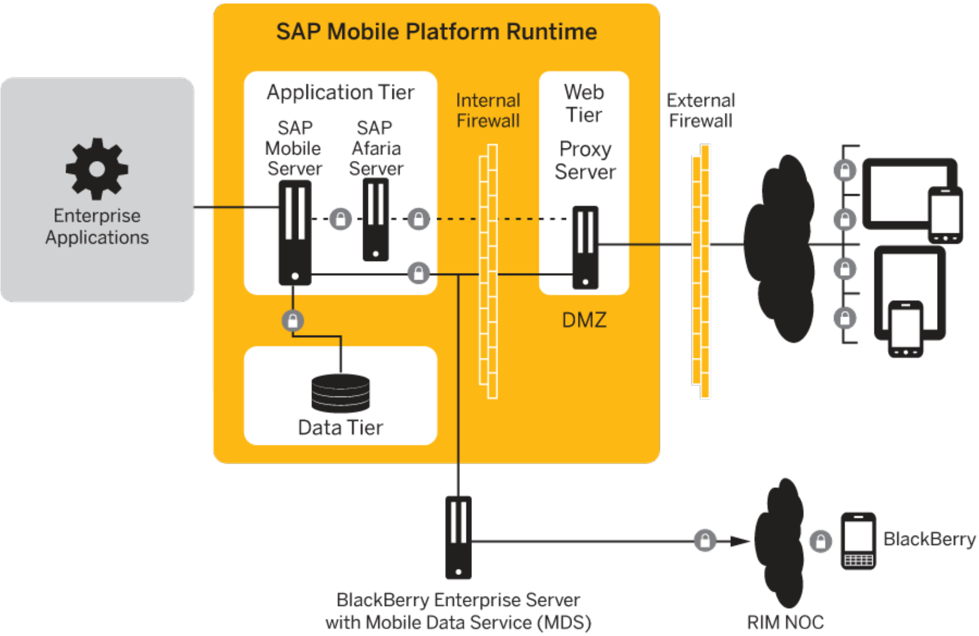
Learn about developing this type of application for deployment to SAP Mobile Platform with *Developer Guide: REST API Applications*.

SAP Mobile Platform Runtime

SAP Mobile Platform Runtime provides the platform infrastructure that allows you to deploy and manage your mobile applications.

The platform, as mobile middleware, serves as the information bridge between device users and enterprise data which is secured behind the corporate firewall or hosted in a cloud infrastructure.

This platform infrastructure is installed alongside other corporate assets in a tiered architecture. SAP Mobile Platform can be installed as a clustered system, which allows conventional load-balancing and failover mechanisms to provide higher system availability.



The client tier consists of device applications built using the SAP Mobile SDK or supported 3rd Party SDK and provisioned to devices.

- *Web Tier*

  The Web tier includes a proxy server, which enables secure, load-balanced communication between mobile devices and SAP Mobile Server. Across-the-firewall deployment occurs without opening any internal firewall ports for enterprise mobilization and ensures that data is secure while in transit. Relay Server or a third-party proxy server, such as Apache Reverse Proxy, can be used as the proxy server.

- *Application Tier*

  As the key component of the application tier, SAP Mobile Server handles enterprise data source and application access, communication between the EIS data source and the mobile device, security, transaction processing, and scheduling.
- *Data Tier*

  The data tier stores data used by SAP Mobile Server and the management console within the runtime landscape.
- *Security*

  An SAP Mobile Platform deployment introduces a multilayer approach to corporate security designed for mobility.
- *Supportability*

  SAP Mobile Platform integrates supportability tools, including SAP® Solution Manager, SAP Change & Transport System (CTS/CTS+), and HP LoadRunner.
- *Mobile Application Life Cycle*

  The mobile application provisioning life cycle defines the process by which an enterprise gives users access to applications and other related mobility resources (enterprise data, network connections, and so on) allowing the application to mediate the user interaction with your organization in a sustained and controllable fashion.
- *Administration and Monitoring*

  Administrators interact with SAP Mobile Platform primarily to configure platform components and ensure the production environment works efficiently as a result of that configuration. A Web-based administrative console, SAP Control Center, is used to manage and administer SAP Mobile Servers, device application deployment, and mobile devices running across the corporate network.

Web Tier

The Web tier includes a proxy server, which enables secure, load-balanced communication between mobile devices and SAP Mobile Server. Across-the-firewall deployment occurs without opening any internal firewall ports for enterprise mobilization and ensures that data is secure while in transit. Relay Server or a third-party proxy server, such as Apache Reverse Proxy, can be used as the proxy server.

The proxy server is used in the runtime architecture as a component of the enterprise demilitarized zone (DMZ), to securely integrate mobile devices into your system landscape. You need not install any SAP Mobile Platform components on the carrier network or outside your firewall; they can all be installed and controlled within your corporate networks.

The proxy server:

- Provides a single point of contact for devices and is a specialized reverse proxy that avoids opening inbound ports in the firewall to SAP Mobile Server.

- Accepts and forwards requests from remote clients to SAP Mobile Platform components.
- Integrates with existing security for Web and enterprise infrastructure, eliminating the need for changes to existing corporate firewalls and IT policies.

SAP Mobile Platform synchronization uses Relay Server end-to-end encryption and HTTPS to secure communication. Devices synchronize with SAP Mobile Platform securely on the device client side.

In an implementation targeting BlackBerry devices where a BlackBerry Enterprise Server (BES) is installed, Relay Server can be optional. In this landscape variation:

- BES acts as the reverse proxy server managing secure access to the network and connections to devices through the RIM Network Operations Center (NOC). However, if BES does not act as the reverse proxy server, Relay Server installed between BES and SAP Mobile Server can perform this function.
- A component of the BES can also manage load balancing. However, if this component is not used for load balancing purposes, Relay Server can be installed as the load balancer between the BES and SAP Mobile Server.

For additional information see:

- *Landscape Design and Integration*
- *Administer > Relay Server* in *SAP Control Center for SAP Mobile Platform*.

Application Tier

As the key component of the application tier, SAP Mobile Server handles enterprise data source and application access, communication between the EIS data source and the mobile device, security, transaction processing, and scheduling.

SAP Mobile Server supports:

- Optimized access to back-end systems, messaging, security services, multi-tenancy capabilities, and monitoring and development activities.
- Data delivery to and from device applications by applying transactions to the cache database, and propagating transactions to back-end systems respectively.

After developing and packaging a mobile application, the package is deployed to SAP Mobile Server. From SAP Mobile Server, the device application can then be deployed to a mobile device, or provisioned to multiple devices using Afaria® Frontline Management, available as a separately licensed and installed product.

The SAP Mobile Server Customization and Management APIs support advanced requirements of complex data handling, transaction execution, security customization, exception handling, and systems management automation, all of which are intrinsic to complex corporate and multi-tenant-based mobile application deployment environments.

The following sections describe key aspects of the runtime landscape. For more detailed information, see these documents:

- *Landscape Design and Integration*
- *Installation Guide for Runtime*
- *System Administration*
- *Security*

Data Tier

The data tier stores data used by SAP Mobile Server and the management console within the runtime landscape.

Four categories of data are persisted and managed in the data tier:

- Cache data is stored only for applications that use synchronization and maintains records of device synchronization activity along with any cached data that may have been retrieved from the back end or pushed from the device.

  The application package hosted in SAP Mobile Server communicates to the CDB through JDBC connection pools that are configured from the administration console. The MBO parameters and the relationships between MBOs define the shape of the cache tables. The internal implementation of these tables and the associated queries are not public and may change from release to release.
- Cluster data maintains knowledge of and coordination between SAP Mobile Server clusters.
- Monitoring and domain log data stores monitoring, tracking, and log messages for application activities.
- For applications that employ messaging services, HTML5/JS Hybrid Apps, OData SDK and applications that use notification mechanisms, this data stores in-transit asynchronous messages until they are processed by the mobile application infrastructure layers. Message data is encrypted on the device and in transit.

For additional information see:

- *Landscape Design and Integration*
- *Installation Guide for Runtime*
- *Data Tier Administration Overview* in *System Administration*

Security

An SAP Mobile Platform deployment introduces a multilayer approach to corporate security designed for mobility.

This approach ensures that:

- Internal and external device users can securely connect to enterprise information systems.
- Every network link that transfers corporate information and every location that stores enterprise data guarantees confidentiality.

SAP Mobile Platform security features cover:

- **Component Security –** SAP Mobile Platform consists of multiple components that are installed on internal networks, primarily on the corporate LAN and the demilitarized zone (DMZ). Each component requires specific administration tasks to secure it.
- **Communication Security –** Secure SAP Mobile Platform component communications to prevent packet sniffing or data tampering. Different combinations of components communicate with different protocols and different ports.

  End-to-end data encryption support is based on Transport Layer Security (TLS) and Secure Sockets Layer (SSL), which secures client/server communication using digital certificates and public-key cryptography.
- **Authentication and Access Security –** Authentication and access control is a core feature supported by all application types to control access to enterprise digital assets.

Security applies to components of the runtime landscape. See *Security* for additional detail for the following.

- **Server Security –** The SAP Mobile Server provides data services to device clients by interacting with the data tier on behalf of the client. The data tier is installed with the server tier components to the internal corporate LAN. Communications with device clients are routed through an open port on the internal firewall to the Relay Server.

  Each runtime service uses its own communication port (secured and unsecured). Security for this tier secures both the server components that provide these services and service communications.
- **Data Tier Security –** The data tier consists of multiple databases that need protection. Each database contains sensitive enterprise data. This level of security involves securing the data infrastructure, then securing the databases by managing DBA passwords, granting DBA permissions, as well as encrypting data and logs.
- **DMZ Security –** DMZ security involves controlling internet traffic to private networks. Key to this control is a proxy server, Relay Server or a third-party proxy server, which resides between inner and outer firewalls.
- **Device Security –** Multiple mechanisms can be combined to fully secure devices. In addition to using the built-in security features of both the device or SAP Mobile Platform, SAP recommends that you also use Afaria so security features can be remotely initiated as required.

  Key SAP Mobile Platform security features for devices include the encryption of data, the implementation of login screens, and the use of DataVault to store sensitive data.

  Application security is based mainly on the mapping of a mobile business object (MBO) package to a security configuration. A security configuration defines the authentication, authorization, attribution, and auditing security provider for an application package's access control and activities. For example, for an application, an administrator may create a security configuration that points to the LDAP server for authentication and authorization, and does not associate any provider for attribution and auditing.

Single sign-on (SSO) enables mobile device application users to enter credentials only once to gain access to all resources, including servers, packages, and data sources related to that application. SAP Mobile Platform supports SSO authentication for mobile clients that access data from an SAP enterprise information system (EIS) using either X.509 certificates or SSO logon tickets (SSO2). In addition, administrators can use their SSO system of choice with SAP Mobile Platform to achieve end-to-end integration across client applications and Enterprise Information Systems (EIS) resources.In addition to supporting X.509 certificate security, SAP Mobile Platform expands single sign-on support to third-party and standard single sign-on mechanisms. With expanded single sign-on support, SAP Mobile Platform enables the authentication framework to accept HTTP headers and cookies propagated by the client or a proxy server and then authenticate and propagate the user to the EIS.

SAP Mobile Platform supports Afaria device management and security functionality. Client applications can generate certificate requests which in turn are passed through Afaria to the corporate PKI system for CA signature. If Afaria is not deployed, the process for generating and provisioning client certificates follows the standard corporate certificate request and renewal process. Afaria device management and security functionality includes features such as remote device locking, remote data cleanup, data fading (a feature that enables the IT administrator to lock, wipe, or reset a device that has not communicated with the corporate network or Afaria server after a predetermined number of days), and password expiration management. Even without Afaria, the SAP Mobile Server administrator can lock or unlock devices from accessing applications deployed to the server.

- **EIS Security –** Secure interactions with EIS data sources. If you are using DCN, those notifications can be communicated to SAP Mobile Server securely.

See *Security* for additional information.

The SAP Mobile Platform Common Security Infrastructure (CSI) provides an extensible model for integrating with existing security infrastructure. CSI login modules conform with Java JAAS, which enables SAP Mobile Platform to integrate with LDAP, Microsoft AD, SiteMinder, etc. For additional information about developing a custom authentication or authorization provider, see *Security API* in *Developer Guide: SAP Mobile Server Runtime*.

Supportability

SAP Mobile Platform integrates supportability tools, including SAP® Solution Manager, SAP Change & Transport System (CTS/CTS+), and HP LoadRunner.

SAP Solution Manager

SAP Mobile Platform is integrated with SAP Solution Manager to support end-to-end tracing and change analysis, as well as workload analysis.

- **SAP Solution Manager integration –** once you configure SAP Solution Manager to recognize SAP Mobile Platform, use SAP Control Center to configure the Solution Manager URL. Clients upload end-to-end trace data to SAP Mobile Server, which uploads it to the Solution Manager to which the URL points. Solution Manager allows you to open SAP Control Center where you can review the End to End trace records from the server. This helps you to get a full trace of a request from client to server.
- **Perform end-to-end trace analysis for Object API, OData, and Hybrid Apps –** application developers use end-to-end trace APIs provided with the corresponding SDK (Object API, OData, and Hybrid Apps) to build trace capability in the client application. The end user initiates tracing in the application when there is a problem, and performs the action that reproduces the problem. The trace data is collected and uploaded from the application to SAP Mobile Server and then sent to Solution Manager. The administrator or help desk staff uses the Solution Manager user interface to analyze the uploaded trace data, which traces activity from device to enterprise information system (EIS), to isolate the root cause of performance and functional problems. For on-device application performance, developers use APIs to enable collection of the performance counters for supported key performance indicators (KPIs), and uses SAP Control Center to view the collected performance data.
- **Perform end-to-end workload analysis for Object API, OData, and Hybrid Apps –** the administrator configures the Introscope agent on the SAP Mobile Server node. The agent sends KPIs, which can be reviewed using Solution Manager or the Introscope Enterprise Manager dashboard user interface.
- **Perform end to end change analysis –** the administrator can record all changes in a solution (for example, operating system, database, Java parameters, transport requests, Support Packages). Solution Manager is the central entry point for root cause analysis, providing number of changes per system, change category, and day.
- **Perform end to end exception analysis –** Solution Manager provides exception trend analysis and a starting point for analysis of a specific exception using the SAP Mobile Platform server log, domain log and Windows event log.
- **System monitoring and alerts –** Solution Manager features a unified alert inbox as a central access point to handle all types of alerts, which enables efficient alert handling based on consolidation of single alerts to alert groups. It integrates the most common alert handling mechanisms, such as: status tracking, incidents, notifications and third-party integration. Administrators can drill down from alert type to alert groups, alert instances

and single metrics and events, as well as view integration of analysis capabilities as problem context and monitoring applications. A status overview allows Administrators to access landscape information and problem context for technical systems with drill-down capability from status information to single metrics and events provided by End-to-End Monitoring and Alerting. Visualization of metrics and events, including thresholds and current rating / value are provided, including jump-in capability in the metric viewer with zoom functionality.

SAP Change & Transport System Integration

SAP Mobile Platform is integrated with Change & Transport System (CTS/CTS+) to enable transport of application, MBO package, and Hybrid App package definitions from development to test and production landscape environments. The SAP Control Center provides export/import of applications, MBO packages, and Hybrid Apps to and from the corresponding archive.

HP LoadRunner Integration

SAP Mobile Platform provides the hooks to integrate with your HP LoadRunner installation (separate license required), which enables the performance architect or developer to perform load and volume testing using standard tools that are available to SAP customers. A usage example for the tool includes load testing SAP Mobile Server with clients using Hybrid Web Container, and clients using OData over an HTTP channel.

See the following for additional information about Supportability:

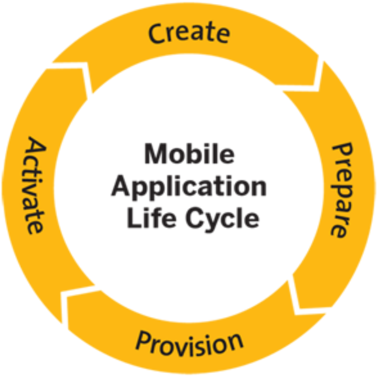
- *System Administration*
- *SAP Control Center for SAP Mobile Platform*
- *Developer Guide: OData SDK*
- *Developer Guide: <Device Platform> Object API Applications*

See also

- *Implement Three Environments* on page 34
- *Object API Application Development* on page 36
- *HTML5/JS Hybrid App API Development* on page 38
- *OData Application Development* on page 41

Mobile Application Life Cycle

The mobile application provisioning life cycle defines the process by which an enterprise gives users access to applications and other related mobility resources (enterprise data, network connections, and so on) allowing the application to mediate the user interaction with your organization in a sustained and controllable fashion.



- **Create –** The create stage consists primarily of package and application development, using SAP Mobile Platform-supported mobile data models and SAP Mobile SDK, or a 3rd party SDK and the SAP Mobile Platform REST API.
- **Deploy –** For applications that use mobile business objects, MBO packages are deployed to SAP Mobile Server to make a package available for a selected domain. It is either a development or administrative task, depending on whether you are in a preproduction or production environment.

  Once deployed, a package can be accessed and used by an application or managed and secured by a domain administrator. The application connection determines how the package engages and interacts with the SAP Mobile Platform runtime.
- **Provision –** Application provisioning is the process of moving the client application to the device. There are four key steps to application provisioning:

  1. Copying the application to the device.
  2. Installing the application on the device.
  3. Configuring the application.
  4. Securing the device.

  In the development test environment, client application binaries and their dependencies are provisioned to emulators or actual devices. Deployment to devices can be achieved by physically connecting the device to the developer's machine, and copying binaries using device-supported tools. Client applications should be tested in a test environment that emulates the production environment for complete end-to-end testing.

  For production scale deployment, application provisioning is managed using proven products such as Afaria. Afaria offers enterprise-class device application deployment and management features.
- **Activate –** Application activation -- part of the user onboarding process -- is complete when the application/user starts interacting with the system. From the application user standpoint, this is the automated series of events by which a user, without administrative

intervention, can start quickly consuming services. This is actually a multi-step process coordinated by the administrator.

Connection registration happens via SAP Control Center by defining the conection and setting connection properties. This process registers connections with SAP Mobile Server using the manual or automatic method and pairs the user, device, and application. When this pairing is complete, the application connections are activated.

Following connection activation, the connection registration pairing is used to determine which connections and customizations are assigned. This assignment activates the application, allowing the user to interact with the system.

- **Maintain –** Maintenance is about keeping your mobility environment functional and mobile operations healthy. The result of maintenance is system reliability and the preservation and prolonging the life of mobility assets

See *Mobile Application Life Cycle* for comprehensive information about these stages.

Afaria Server is a recommended component of the landscape to provide enterprise level device application deployment and device management. With SAP Mobile Server, the solution supports end-to-end management and security of devices, and over-the-air (OTA) and push-based deployment of device applications.

Afaria extends SAP Mobile Platform functionality by providing additional device client management features for remote and mobile computing devices, including laptops, desktops, and handheld devices. When Afaria is licensed and implemented with SAP Mobile Platform, the Afaria management console is launched from SAP Control Center.

Use the Afaria tools to:

- **Deploy client and runtime infrastructure components –** Using OTA and push-based deployment, mobile devices automatically receive notifications, application updates, etc. Afaria administrators can add, update or remove applications, data and content without the users' involvement and can ensure mobile workers have the correct software and data in the field. Users can be confident that the data on their devices is up-to-date and reliable.
- **Track assets –** Identify and track assets being used by your mobile workforce. Afaria Administrators have full control of the range of devices and applications deployed and can view information from a single console.
- **Secure devices and data –** Data and content is backed up and can be deleted if a device is lost or stolen.

Administration and Monitoring

Administrators interact with SAP Mobile Platform primarily to configure platform components and ensure the production environment works efficiently as a result of that configuration. A Web-based administrative console, SAP Control Center, is used to manage and administer SAP Mobile Servers, device application deployment, and mobile devices running across the corporate network.

There are two types of administration involved in management of the runtime environment.

- SAP Mobile Platform administration (also known as the platform administrator) – for the administrator managing the whole environment, this type has cluster-wide administration access to the SAP Mobile Platform. The SAP Mobile Platform administrator manages the lifecycle of the domains including users who have access to it.
- Domain administration – providing access confined to the specific domains that the platform administrator assigns, this type is for the customers of the SAP Mobile Server hosting provider where they can perform package management, subscription management, log viewing, role mapping and server connections management operations in one or more domains assigned to them. The domain forms the logical container and unit of separation of the artifacts contained in it.

SAP Control Center allows administrators to centrally manage, secure, and deploy servers, data applications and devices.

This console enables administrators to view server status; perform start, stop, restart operations; configure server settings; deploy packages; configure package settings (including cache group schedule, and synchronization settings); manage subscriptions (including unsubscribe, recover, suspend, and resume); set push notifications; perform role mapping, cluster management tasks, and device lifecycle and licensing tasks (including registration, unregistration, locking, unlocking, among others).

SAP Control Center assists with:

- Core management activities
  - **Server configuration –** Configure server parameters including performance, network ports, messaging, SSL security, and log levels among others.
  - **Cluster management –** Register and monitor clusters to ensure that SAP Control Center clusters and servers work smoothly, and scale over time. When the SAP Mobile Platform deployment is part of a larger SAP landscape, System Landscape Directory (SLD) server and SAP License Audit server support are provided to assist in keeping information about your SAP and SAP Mobile Platform mobility infrastructure complete and current.
  - **Connection registration and security –** Facilitate the application activation process. Control and monitor devices registered. Registration, activation, and provisioning of devices allows device users to access applications and to be tracked and managed by the system.
  - **Security configuration, administration, and application –** Configure and manage permissions that allow execution of operations on registered servers.
  - **Monitoring –** View historical application usage data, view performance statistics, and manage monitoring data.

    The monitoring component is where the platform administrator can view synchronization activities, user access activities, key performance indicators for various activities in the server, cache performance, and other data being gathered. The administrator can set the monitoring configuration to enable/disable monitoring on

selected domains and packages, and also perform monitoring data cleanup and export it to share with others for reporting and auditing purposes.
  - **Multitenancy management –** Deploy a single production environment to service multiple client organizations, known as Tenants. Administrators can view domains that are created to support the tenancy strategy.
  - **Log viewing –** Review logs directly from the console.
  - **Troubleshooting –** Use information gathered by monitoring SAP Mobile Platform components to identify and troubleshoot issues.
- Application management activities
  - **Device application deployment –** Manage and monitor device applications deployed to SAP Mobile Server. SAP Afaria Frontline Management is required.
- Package management activities
  - **Subscription management –** Configure, manage, and monitor subscriptions related to packages and device user profiles. Subscriptions define what subset of data the device user wants to receive when the device is synchronized with the server.
- Proxy Connection management activities: Manage the endpoint for the proxy connection

In addition to the console, the systems management framework can integrate with Simple Network Management Protocol (SNMP) management tools to monitor specific server events. You can use the SAP Mobile Server management API to automate and extend administration, management, and monitoring features.

For additional information see:

- *SAP Control Center for SAP Mobile Platform*.

See *System Administration* for information about the following administration areas:

- Cluster administration
- Server administration
- Data tier administration
- EIS connection management
- Domain management
- Performance tuning
- Backup and recovery
- Platform monitoring and diagnostics
- SAP interoperability

Index