SYBASE®
An SAP® Company

Fundamentals

Sybase Unwired Platform 2.1
ESD #2

DOCUMENT ID: DC01204-01-0212-01
LAST REVISED: February 2012
Copyright © 2012 by Sybase, Inc. All rights reserved.
This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.
Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.
Sybase trademarks can be viewed at the Sybase trademarks page at *http://www.sybase.com/detail?id=1011207*. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.
Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.
Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.
All other company and product names mentioned may be trademarks of the respective companies with which they are associated.
Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.
Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

CHAPTER 1: Enterprise Mobility and Mobile Application Usage Scenarios 1

Enterprise Mobility Requirements 3
Device User Requirements 4

CHAPTER 2: Platform Overview 5

CHAPTER 3: Mobile Data Models for Mobile Application Development 7

Mobile Business Objects 8
Data Modeling 9
Mobility Patterns 9
MBO Attributes 10
Operations 10
Relationships 10
Other Key Concepts 11
Data Caching 11
Object API Code Generation 12
Package Deployment 13
OData 13
Data Orchestration Engine 14

CHAPTER 4: Sybase Mobile SDK 15

Development Life Cycle 17
Life Cycle Stages 17
Stage 1: Development 18
Stage 2: Testing 19
Testing in the Development Environment 20

Stage 3: Production 20
Stage 4: Maintenance 21
Best Practices 21
Implement Three Environments 21
Implement Change Control 21
Object API Applications 22
Object API Application Development 23
Building Mobile Analytics into Object API Applications 24
HTML5/JS Hybrid Apps 24
HTML5/JS Hybrid App API Development 25
Building Mobile Analytics into HTML5/JS Hybrid Apps 26
OData SDK Applications 26
OData SDK Development 27

CHAPTER 5: Unwired Platform Runtime 29

Runtime Landscape 31
Unwired Server 31
Data Tier 31
Relay Server 32
Afaria Server 33
Clustered Versus Non-Clustered Systems 34
Security 34
Application Deployment 36
Application Connection Registration 37
System Management 37
Administration and Monitoring 38

CHAPTER 6: Documentation Roadmap 41

Documentation for Developers 42
Mobile Data Models Development Documentation 42
Object API Application Development Documentation ...44
HTML5/JS Hybrid App Development Documentation ...45

OData SDK Application Development 46
Documentation for Administrators 46
Runtime Administration Documentation 46
External Systems Administration Documentation 48
Administration Automation Development Documentation 48
Documentation Roadmap for Unwired Platform 49

Index 55

CHAPTER 1 Enterprise Mobility and Mobile Application Usage Scenarios

Extending your enterprise business processes, data, and information away from the office entails many technical challenges. To successfully mobilize your enterprise using Sybase® Unwired Platform, determine which mobile application categories fit your enterprise mobility use case and balance device user requirements with enterprise requirements.

Mobility solutions present complexity to the IT organization. These complexities may include, but not be limited to, lack of integrated solutions, heavy upfront costs for large-scale customization requirements, and the lack of repeatable development and deployment models. Sybase Unwired Platform addresses these complexities by providing a platform-based solution that allows you to leverage existing mobile development tools and expertise to develop applications to meet your users' data needs, implement to complement areas of your existing infrastructure and security, and access data from a variety of enterprise information systems and deploy to a variety of mobile platforms.

Sybase Unwired Platform supports development and deployment of several categories of mobile applications. Each category identifies the specific application archetype for development.

- **Offline mobile applications –** With offline mobile applications, information availability is mission critical regardless of connectivity. Application requirements typically involve:
  - Need for some sets of data available on the device for use if the application is used offline.
  - More complex task orientation, data input, and screen navigation

  Offline applications are developed using the Sybase Mobile SDK Object API. Object API applications leverage Unwired Platforms synchronization services, developers create applications that allow users to work offline (create, update data). Data is synchronized between the EIS and the device when connectivity is available.
- **Hybrid Container applications –** Hybrid mobile applications offer an option that is a hybrid of HTML5 and container 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 user experience, involving notifications, lookup, and action-decision forms

- No long-lasting offline stateful transactions

Hybrid Container applications leverage Unwired Platforms services from within an standards based HTML(5)/JavaScript environment, developers create applications using Sybase Mobile Workflow Designer or using third party web tools. With this approach, a container is developed natively for the target mobile operating system and deployed to the device, then one or more workflows are deployed to the container.

- **OData SDK mobile applications –** With OData 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 Sybase Mobile OData SDK. OData SDK applications leverage Unwired Platforms Online Data Proxy services. Developers create applications that allow users to work (create, update data), connecting the application to SAP® NetWeaver Gateway exposed data.

Mobile applications developed and deployed with Sybase Unwired Platform address common services, covering the following areas:

- Device, data, and transport security
- Error reporting and troubleshooting support
- Remote device management support

Sybase Unwired Platform, by providing the development tools to support these mobile application types and by providing the runtime platform that deploys and manages mobile applications, meets enterprise mobility requirements and device user expectations requiring development skills found in most organizations.

See also

- *Chapter 4, Sybase Mobile SDK* on page 15
- *Object API Applications* on page 22
- *OData SDK Applications* on page 26
- *HTML5/JS Hybrid Apps* on page 24

Enterprise Mobility Requirements

Sybase Unwired Platform, with Sybase Mobile SDK and Unwired 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 Sybase Unwired Platform support common device user requirements. Support of these user requirements are balanced with enterprise requirements.

Common device user requirements may include:

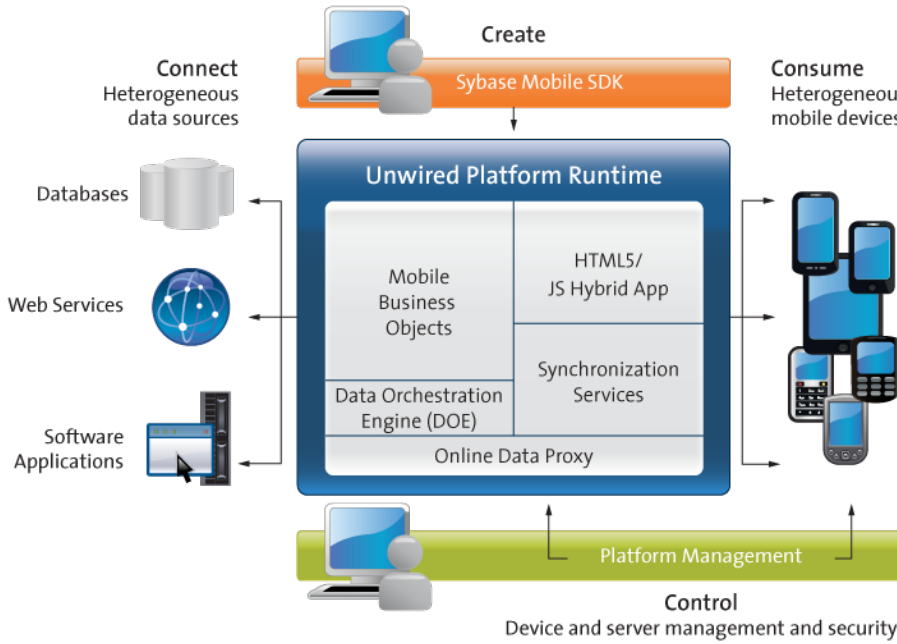
- Device choice. More organizations now support this as well as the ability to configure a personal device for work productivity and vice-versa.
- Ease of use. Devices and applications that are easy to configure and use. App interfaces must be intuitive and integrate with device features, such as automatic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

CHAPTER 2 Platform Overview

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

Sybase Unwired Platform acts as the hub that connects the back-end enterprise systems and data sources to mobile devices. Features for mobile application development, deployment, security, and ongoing device and application management provide a complete end-to-end solution. The platform is made up of:

- **Sybase Mobile SDK –** The platform development tool set used to build mobile applications that meet your mobility needs.
- **Sybase Unwired Platform Runtime –** The deployment and management architecture and services used to run and manage mobile applications.



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 Sybase Mobile SDK to build and test mobile applications that meet your mobility needs.

- **Control –** Deploy to and manage Unwired 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.

- *Mobile Data Models for Mobile Application Development*

  There are three ways to model enterprise data so that mobile applications can be developed to access data in your enterprise: Mobile Business Objects using the Sybase Mobile SDK, Data Objects using SAP Data Orchestration Engine (DOE), and OData using SAP® NetWeaver Gateway.
- *Sybase Mobile SDK*

  Sybase Mobile SDK supports the development lifecycle for Object API Applications, HTML5/JS Hybrid Apps, and OData SDK Applications.
- *Unwired Platform Runtime*

  Unwired 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 Sybase® Unwired Platform and determine where to start based on your role.

CHAPTER 3 Mobile Data Models for Mobile Application Development

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

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	Mobile Business Object or Data Orchestration Engine
Hybrid Container applications	HTML5 / JS hybrid apps	Mobile Business Object
Online applications	OData SDK applications	OData

Based on your data model and business logic, the EIS interface must be defined. This process involves eventually involves adapting existing services exposing data from the EIS. Your data models will 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 Unwired WorkSpace graphical tools, which are provided with Sybase 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 Sybase Unwired Platform messaging.

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 Unwired WorkSpace graphical tools, which are provided with Sybase 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 Sybase Unwired Platform, MBOs provide a layer of abstraction from Unwired Server's interaction with heterogenous back ends/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*
- *Sybase Unwired WorkSpace: Mobile Business Object Development*

See also

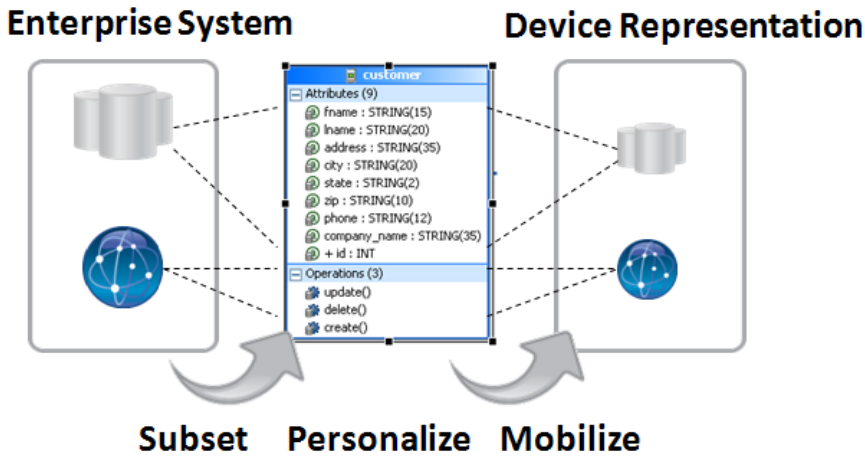
- *Object API Applications* on page 22
- *HTML5/JS Hybrid Apps* on page 24

Data Modeling

After connecting to back-end data sources using connection profiles from Unwired 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 Unwired Server where the MBO manages synchronization between the EIS and Unwired 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.



Mobility Patterns

MBO design allows developers to build in support for mobility patterns—interactions between the EIS, the MBO, and the device cache. These patterns include data virtualization and operation replay.

- Virtualization – normalizes the data and semantics for interacting with different enterprise information systems (EISs), each with its own set of connection interface, data, operation, and type structures.

  An MBO provides a layer between your enterprise servers or applications, and the remote database on the device client. Data virtualization utilizes a cache database–also referred to as the CDB or Unwired Server cache–to optimize device client access and minimize back-end resource utilization.

- Operation replay – supports execution of the client-initiated transactions (for example Create, Update, and Delete operations) that result in data changes on the EIS.

MBO Attributes

MBO attributes define the structure for the data associated with the MBO instance on the mobile device.

Attributes define the scope of the device-side data store. Attributes and synchronization parameters in an MBO define the server cache. The server cache and device data store are populated by reading data from the EIS using exposed services or standard protocols and methods, such as a SQL select statement for a database data source.

Attributes also have additional metadata provided during its Read/Load operation, such as specification of a load argument and synchronization parameter option that can be used to provision user-specific data, i.e., the load arguments narrow down the data downloaded to the CDB, and the synchronization parameter filter the CDB data further when downloading to the mobile device. For example, a sales representative in the Eastern region may be interested only in seeing data for that region. Developers can build the application to map those preferences to MBOs to drive the data filtering specific to the user. The same data obtained from the EIS is then further partitioned and used to serve all users, thereby optimizing requests for data to the EIS and improving performance of the mobile application.

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.
- Read/Load – an operation that inlcudes optional load arguments that determine inital loading of data from the EIS to the CDB. For example, a SQL Select statement for database data sources.
- 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.

Relationships

Relationships define the data association between two MBOs by linking attributes and load arguments (the read operation's parameters) in one MBO to attributes and load arguments in another MBO.

Relationships help provision related data as one unit, and properly sequence the operations on the related MBOs based on real-time detection of the object instances used. Relationships can be one-to-many, many-to-one, or one-to-one.

From an attribute standpoint, bidirectional relationships are supported; and from an operation execution standpoint, composite relationships are supported.

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, i.e. 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 Unwired 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—can be used to perform single sign-on from the device application to the Unwired Server, authentication and authorization on Unwired 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 Unwired 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, Unwired 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.
- Partitioning data loading – only user relevant data, or partition, is loaded, based on MBO load arguments.
- On-demand loading – data is not filled, until the client performs synchronization.
- Scheduled loading – data is filled periodically according to a schedule you set.

Data change notification (DCN) facilitates propagation of data changes from the back-end enterprise information system (EIS) to the Unwired 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 data sources to keep the Unwired 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 Unwired 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. Unwired 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 Unwired 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 Unwired Server as a deployment package using the Sybase Unwired WorkSpace MBO tools.

When you deploy MBOs to the Unwired 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 Unwired 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 data sources to production data sources.
- 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 SDK Applications* on page 26

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 Sybase Unwired Platform messaging.

DOE-based applications, working in conjunction with Sybase Unwired 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

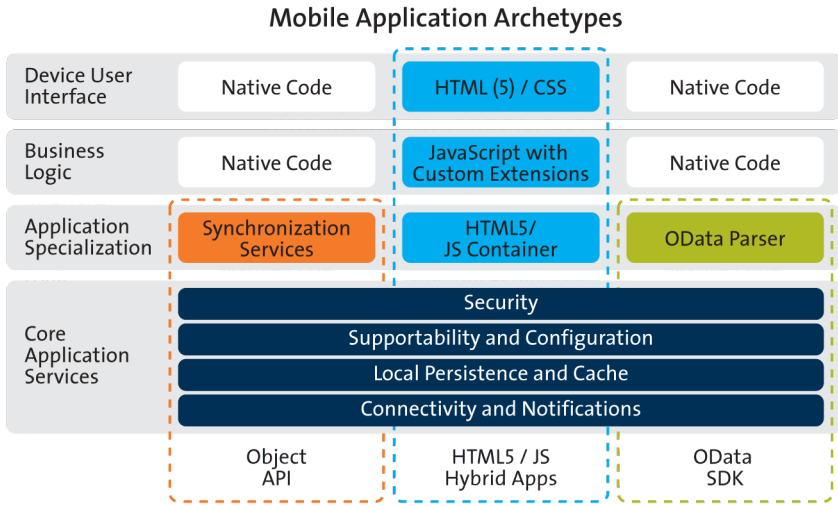
See also

- *Object API Applications* on page 22

CHAPTER 4 Sybase Mobile SDK

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



Dashed outlines in the diagram represent the services provided by Sybase 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 Mobile SDK. Native code development uses Unwired 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 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.

- **Application Specialization –** Application specialization is the differentiator between development approaches. Based on these distinct application 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.
HTML5/JS Hybrid App SDK	Online applications with Push: Simple queries and workflows that follow a request-reply or lookup pattern.
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.

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.

Push leverages a proprietary message transport to move data between device and server. The data change is initiated, or pushed, from where ever 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 Mobile Workflow Forms Editor to design HTML5/JS.

- *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 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 SDK Applications*

  OData SDK 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.

See also

- *Chapter 1, Enterprise Mobility and Mobile Application Usage Scenarios* 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.

Sybase Mobile SDK is intended to work with your existing development paradigm. Use Sybase 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

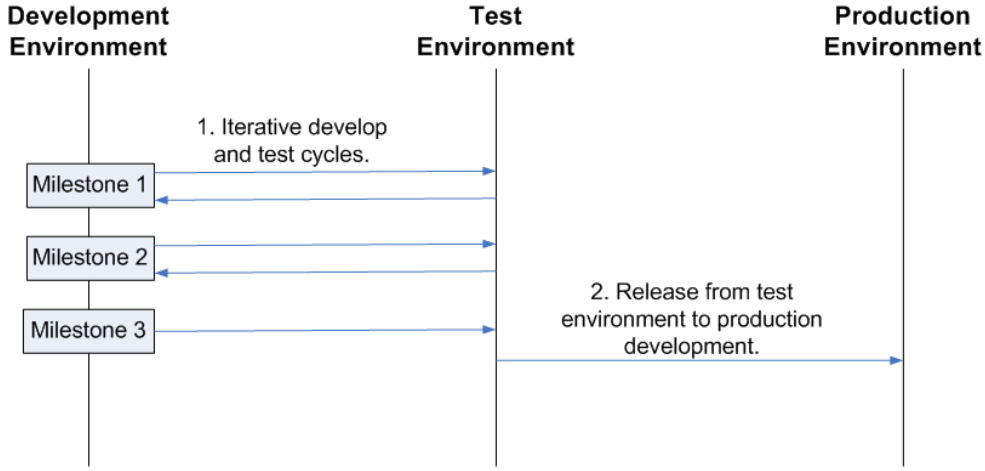
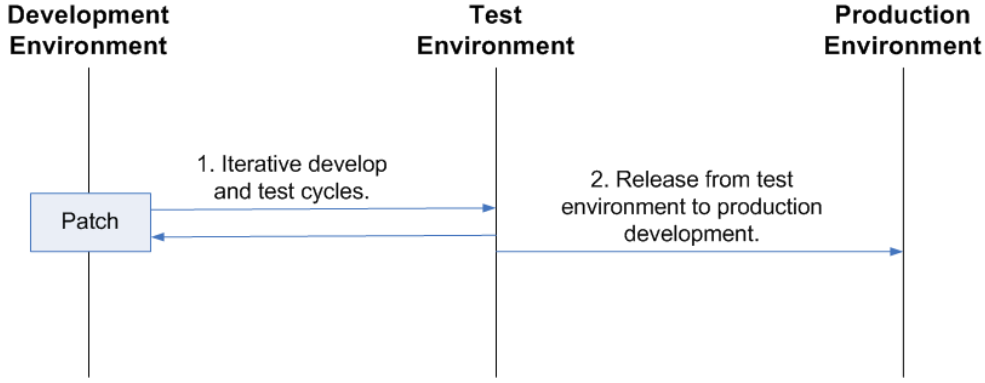


Figure 2: Life Cycle Development for Patches



Stage 1: Development

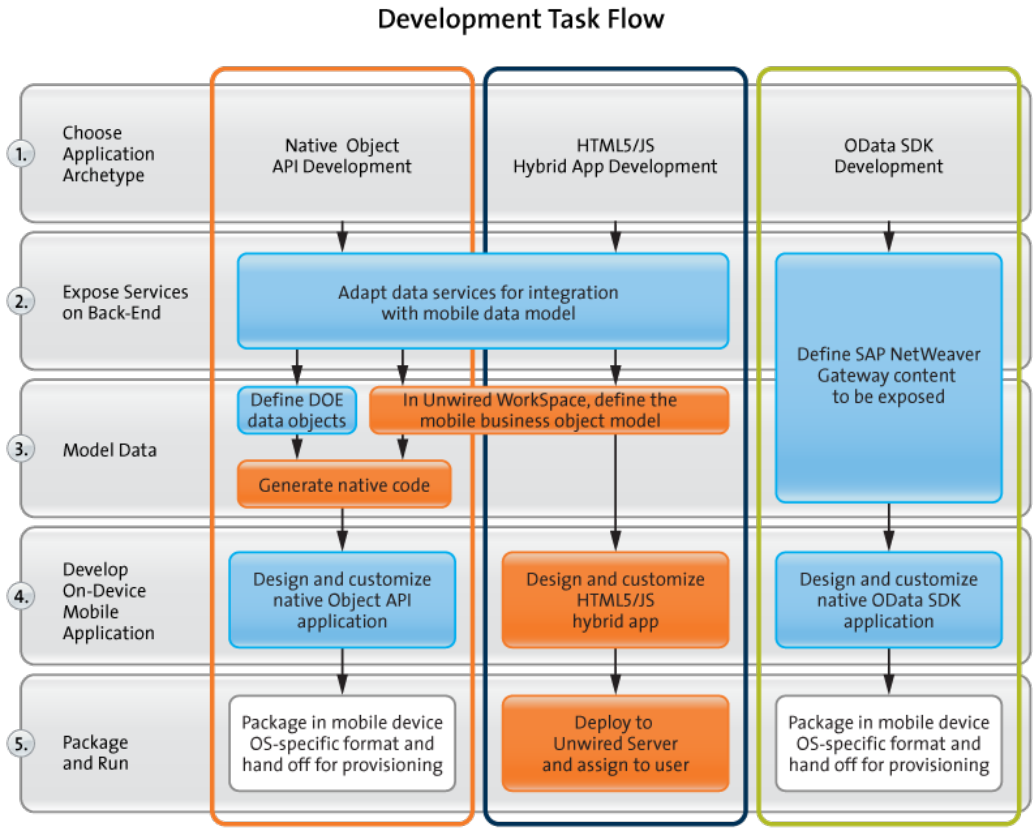
During development, developers use tools to perform requirements analysis, design and model a solution, and develop application code.

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 Unwired 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.



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 Sybase Unwired WorkSpace development environment. Alternatively, you can use Active Sync 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, download the iOS simulator 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 Sybase Unwired 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, Unwired 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.

Sybase Unwired Platform does not provide specific tools to follow the data path through the system. Use an administrative console, log files, and error messages to help validate your testing.

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

Implement Change Control

Implement change control as part of your development process. Integrate your change control system with Sybase Unwired 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 Unwired WorkSpace project file)
  - Native application code, and generated and custom client code
  - Unwired 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. Unwired 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 DOE packages, that are deployed to Unwired 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 14
- *Mobile Business Objects* on page 8
- *Chapter 1, Enterprise Mobility and Mobile Application Usage Scenarios* 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 Unwired 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 Unwired 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 C# 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 Sybase Unwired 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 24

Building Mobile Analytics into Object API Applications

The Mobile Analytics Kit (MAKit) component of Sybase 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 2.1 ESD #2*

See also

- *Object API Application Development* on page 23

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 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. Light weight 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 Unwired Server as a data change notification request (server-initiated).

See also

- *Mobile Business Objects* on page 8
- *Chapter 1, Enterprise Mobility and Mobile Application Usage Scenarios* 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 Sybase Unwired Platform, development tools enable both aspects of HTML5/JS Hybrid App development.

Mobile Workflow applications can reference one or more MBOs and can include load parameters, personalization, and error handling.

Once you have developed MBOs and deployed them to Unwired Server, develop device-resident presentation and logic for your Mobile Workflow application using the Mobile Workflow 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 Sybase Unwired WorkSpace, import the project containing MBOs, connect to Unwired Server and deploy the project.
3. Open the Mobile Workflow Forms Editor and design your worklfow.
4. Package the workflow code and prepare the package for deployment.
5. Test the workflow using a simulator/emulator.
6. Deliver the workflow package to the Sybase Unwired Platform Administrator for production testing and provisioning.

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

- *Sybase Unwired WorkSpace - Mobile Workflow Package Development*
- *Developer Guide: Mobile Workflow Packages*

.

See also

- *Building Mobile Analytics into HTML5/JS Hybrid Apps* on page 26

Building Mobile Analytics into HTML5/JS Hybrid Apps

The Mobile Analytics Kit (MAKit) component of Sybase 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 2.1 ESD #2*

See also

- *HTML5/JS Hybrid App API Development* on page 25

OData SDK Applications

OData SDK 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 SDK 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 Unwired 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 SDK 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 SDK 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

- *Chapter 1, Enterprise Mobility and Mobile Application Usage Scenarios* on page 1
- *OData* on page 13

OData SDK Development

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

Instead of being based on data model elements (MBOs), the OData SDK 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 SDK Applications.

The common development task flow for OData SDK Applications 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 Sybase Unwired Platform Administrator for production testing and provisioning.

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

CHAPTER 5 Unwired Platform Runtime

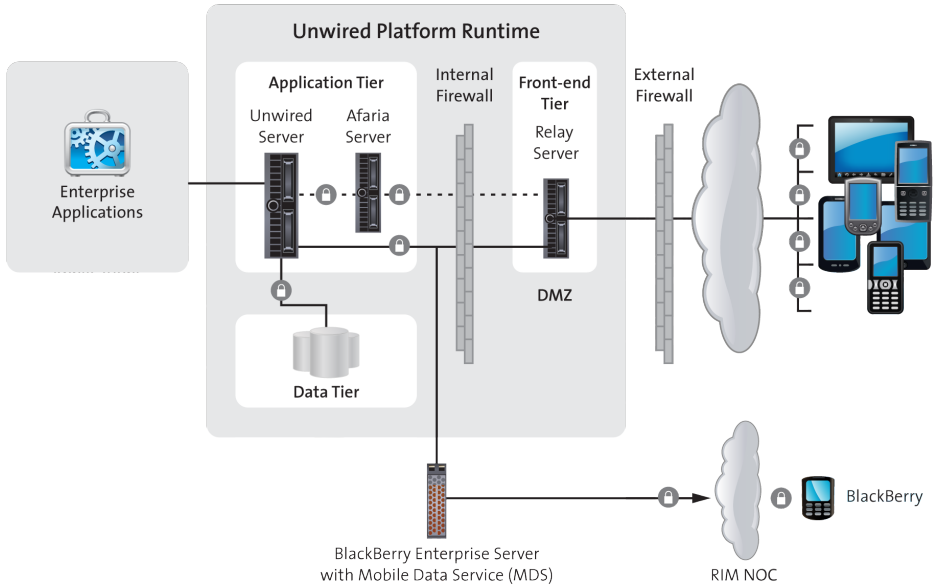
Unwired Platform Runtime provides the platform infrastructure that allows you to deploy and manage your mobile applications.

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 one of two deployment options: Unwired Platform Runtime or Online Data Proxy.

Tiered architecture is common to both options.

- **Application tier –** Integrates the server components with back-end enterprise systems, data access and transaction services, device and application deployment, and system management functionality. Unwired Server and Afaria are part of the server tier, and depending on your deployment strategy may or may not reside on the same server host.
- **Data tier –** Stores data retrieved from the back-end data sources and other runtime related metadata. Multiple databases that assist with monitoring and user agent tracking make up the data tier. For mobile applications that use synchronization and data caching, a separate database is designated in the data tier.
- **Front End tier –** As a plugin to an Apache Web server of Microsoft Internet Information Server (IIS), Relay Server resides in the DMZ to act as the single point of contact for devices and as a specialized reverse proxy that avoids opening inbound ports in the firewall to Unwired Server.
- **Client tier –** Consists of device applications built using the Sybase Mobile SDK and provisioned to devices.



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

- *Installation Guide for Runtime*, System Deployment
- *System Administration*
- *Security*
- *Quick Start: Online Data Proxy*

- *Runtime Landscape*

  The runtime landscape includes multiple components that are part of or interact with Unwired Platform components to provide the platform mobility solution. Installed and configured as a clustered or non-clustered system, Unwired Platform components include: Unwired Server, the Data tier, Relay Server, and Afaria Server.
- *Security*

  An Unwired Platform deployment introduces a multilayer approach to corporate security designed for mobility.
- *Application Deployment*

  During development the developer tests the device application by deploying it to Unwired Platform Runtime. In a production environment, device applications are provisioned using Afaria or similar third-party product.

- *System Management*
  Sybase Control Center for Sybase Unwired Platform offers integrated, Web-based access to all Unwired Servers running in the corporate network.

Runtime Landscape

The runtime landscape includes multiple components that are part of or interact with Unwired Platform components to provide the platform mobility solution. Installed and configured as a clustered or non-clustered system, Unwired Platform components include: Unwired Server, the Data tier, Relay Server, and Afaria Server.

Unwired Server

As the runtime server, Unwired Server handles enterprise data source and application access, communication between the EIS data source and the mobile device, security, transaction processing, and scheduling.

Unwired 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 Unwired Server. From Unwired 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 Unwired 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.

For additional information see:

- *Deployment Planning* in *Installation Guide for Runtime*
- *Server Administration Overview* in *System Administration*
- *Administer: Unwired Server* in *Sybase Control Center for Sybase Unwired Platform* or in *Sybase Control Center for Online Data Proxy*
- *Developer Guide: Unwired Server Management APIs*

Data Tier

The data tier comprises multiple databases installed for and used by Unwired Server and the management console within the runtime landscape.

Other than planning for installation of the data tier, among other deployment planning activities covered in the *Runtime Installation Guide*, and typical DBA maintenance, Unwired

Platform administrators have limited involvement with the data tier. It is important to understand the purpose of each of the databases that make up the data tier.

- **Cache Database (CDB) –** Installed only with the Unwired Platform Runtime option and used only for applications that use synchronization, this relational database management system 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 Unwired 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 Database –** Maintains knowledge of and coordination between Unwired Server clusters.
- **Monitoring and Domain Log Database –** Stores monitoring, tracking, and log messages for application activities.
- **Messaging Database –** For applications that employ messaging services, HTML5/JS Hybrid Apps, OData SDK and applications that use notification mechanisms, this database 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:

- *System Deployment Overview* in *Installation Guide for Runtime*
- *Data Tier Administration Overview* in *System Administration*

Relay Server

Relay Server enables secure, load-balanced communication between mobile devices and Unwired 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 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 Unwired Platform components on the carrier network or outside your firewall; they can all be installed and controlled within your corporate networks.

Relay Server:

- Provides a single point of contact for devices and is a specialized reverse proxy that avoids opening inbound ports in the firewall to Unwired Server.
- Accepts and forwards requests from remote clients to Unwired Platform components.
- Is implemented as a pair of Web extensions that run in a Web server. Relay Server supports two Web servers: IIS on Windows and Apache on Linux.
- Integrates with existing security for Web and enterprise infrastructure, eliminating the need for changes to existing corporate firewalls and IT policies.

Unwired Platform synchronization uses Relay Server end-to-end encryption and HTTPS to secure communication. Devices synchronize with Unwired 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 Unwired 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 Unwired Server.

For additional information see:

- *Relay Server Deployment* in *Installation Guide for Runtime*
- *Administer: Relay Server* in *Sybase Control Center for Sybase Unwired Platform* or in *Sybase Control Center for Online Data Proxy*

Afaria Server

As a separately licensed and installed component, Afaria Server provides enterprise level device application deployment and device management. Integrated with Unwired 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 Unwired 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 Sybase Unwired Platform, the Afaria management console is launched from the Sybase Control Center console.

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.

Additional detail is provided with Sybase Unwired Platform documentation. See:

- *Device and Application Provisioning Overview* and *Provisioning with Afaria* in *System Administration*
- *Device Security* in *Security*
- Documentation provided with Afaria as cited.

Clustered Versus Non-Clustered Systems

Unwired Platform Runtime can be implemented as a clustered or non-clustered system. Clustering creates redundant Unwired Platform components on your network to provide a highly scalable and available system architecture.

A non-clustered Unwired Platform Runtime system is simpler, less expensive to deploy, and generally easier to maintain. However, a non-clustered system has significant limitations:

- It cannot be scaled by adding or subtracting servers, to adapt to changes in system load or performance requirements.
- It cannot take advantage of conventional load-balancing and failover mechanisms to provide higher system availability.
- The only sure way to increase overall system performance is to upgrade the host system resources (CPU, RAM, etc.).

A clustered system design helps to meet requirements for scalability, higher availability, and overall higher system performance

There are two main advantages of a clustered Unwired Platform system:

- It can be scaled by adding or subtracting servers (nodes in a cluster), to adapt to changes in system load or performance requirements.
- Redundant cluster nodes allow conventional load-balancing and failover mechanisms to provide higher system availability.

For additional information see:

- *Deployment Planning* in *Installation Guide for Runtime*
- *Cluster Administration Overview* in *System Administration*
- *Administer: Clusters* in *Sybase Control Center for Sybase Unwired Platform* or in *Sybase Control Center for Online Data Proxy*

Security

An Unwired 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.

Unwired Platform security features cover:

- **Component Security –** Unwired 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 Unwired 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. All applications, except Workflow, use a DataVault to store secrets, including those used to authenticate the user (for example, certificates).

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

- **Server Security –** The Unwired 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 the 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 Relay 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 Unwired Platform, Sybase recommends that you also use Afaria so security features can be remotely initiated as required.

  Key Unwired Platform security features for devices include the encryption of data, the implementation of login screens, and the use of Data Vault 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) security providers provide an alternative to user name and password authentication. These security providers add support for token and certificate-based authentication, such as X.509 certificates. 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.

Unwired Platform supports Afaria device management and security functionality, which 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 Unwired 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 Unwired Server securely.

Application Deployment

During development the developer tests the device application by deploying it to Unwired Platform Runtime. In a production environment, device applications are provisioned using Afaria or similar third-party product.

There are four key steps to application deployment using Unwired Platform Runtime:

1. For Object API applications or Hybrid web applications, mobile data model packages are deployed to Unwired Server.
2. Mobile applications are provisioned to mobile devices.

   In the development test environment, device 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.

   Device applications should also be tested in a test environment that emulates the production environment for complete end-to-end testing.

   For production scale deployment, application deployment is managed using proven products such as Afaria. Afaria offers enterprise-class device application deployment and management features.
3. Application connections are registered with Unwired Platform using the manual or automatic method.
4. Device users start the mobile application and initiate connection to the system.

Application Connection Registration

Application users and their connections are registered in the Unwired Server. Depending on the nature of the application, the registration mechanism differs.

There are two methods: automatic and manual.

- The automatic method uses Afaria for provisioning and enables registering a user upon successful validation of the credentials.
- The manual method uses a white-listing based approach where the application connection is registered by the administrator and specifies an activation code. Then device application user presents the same activation code when the application initializes to complete the registration.

Application connection registration is used for license enforcement, based on the number of unique devices associated with the registered connections. Multiple applications deployed to the same device accessing the same server result in consuming one device license unit.

For additional information, see *Device and Application Provisioning Overview* in *System Administration*.

System Management

Sybase Control Center for Sybase Unwired Platform offers integrated, Web-based access to all Unwired Servers running in the corporate network.

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).

In addition, the Web console is designed for multi-tenancy in mind to support two views:

- Platform administrator view – for the administrator managing the whole environment.
- Domain administrator view – for the customers of the Unwired 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. The Unwired Server administrator manages the lifecycle of the domains including users who have access to it.

Extending the operational and multi-tenancy capabilities of the platform is the monitoring component of the management console 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.

Administration and Monitoring

Unwired Platform uses a Web-based administrative console to manage and administer Unwired Servers, device application deployment, and mobile devices running across the corporate network.

The administration console, Sybase Control Center, allows administrators to centrally manage, secure, and deploy servers, data applications and devices.

This systems management command and control functionality 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 Unwired Server clusters and servers work smoothly, and scale over time.
  - **Device registration and security –** Facilitate onboarding. 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.
  - **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 Sybase Unwired Platform components to identify and troubleshoot issues.
- Application management activities
  - **Device application deployment –** Manage and monitor device applications deployed to Unwired Server. 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 Unwired Server management API to automate and extend administration, management, and monitoring features.

For additional information see:

- *Sybase Control Center for Sybase Unwired Platform* or *Sybase Control Center for Online Data Proxy*

CHAPTER 6 Documentation Roadmap

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

Bookmark *Sybase Unwired Platform Product Documentation* at *http://sybooks.sybase.com/nav/summary.do?prod=1289* to quickly access the complete set of online documentation.

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

Tutorials, Projects and Articles

Check SDN regularly for tutorials, projects and articles: *http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e*.

Tutorials demonstrate how to use Sybase Unwired 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 Sybase Unwired 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 Sybase Unwired Platform. You can download and run these working applications.

Documentation by Role

The following diagrams and document descriptions assist in determining appropriate documents for review by the various roles within Sybase Unwired Platform.

- *Documentation for Developers*

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

  If you are an administrator, identify documentation based on your administrative focus.
- *Documentation Roadmap for Unwired Platform*

  Learn more about Sybase® Unwired Platform documentation.

Documentation for Developers

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

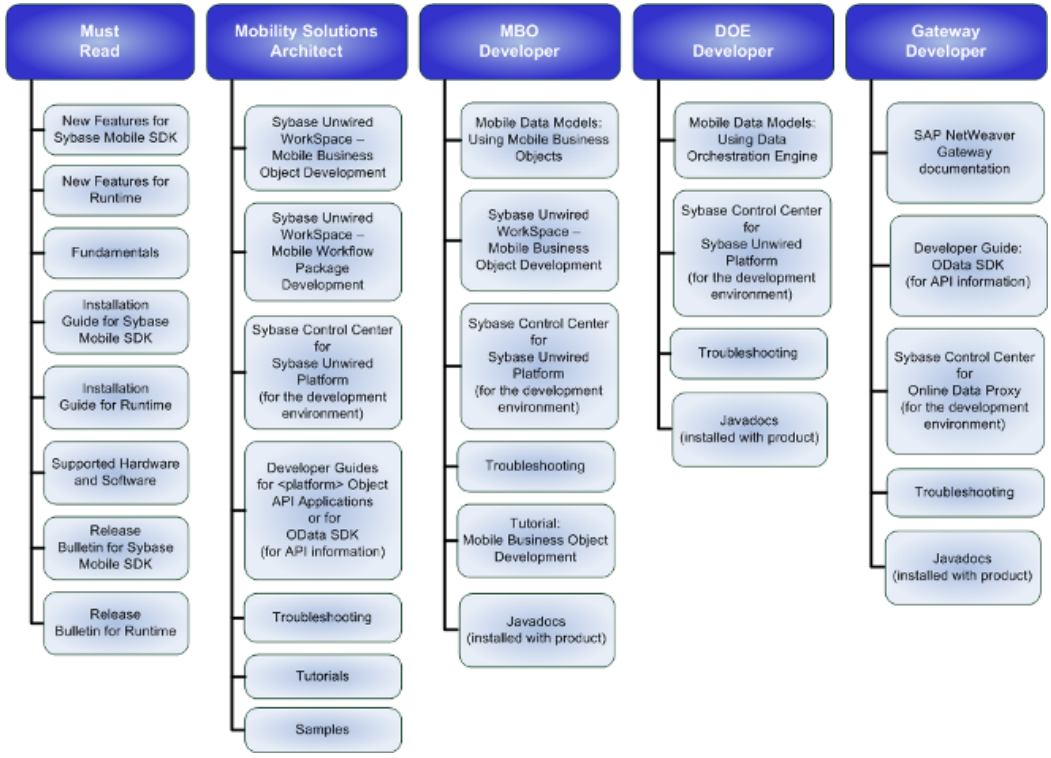
Convenient diagrams are provided to guide you through documentation for the following development categories:

- *Mobile Data Models Development Documentation*
- *Object API Application Development Documentation*
- *HTML5/JS Hybrid App Development Documentation*
- *OData SDK Application Development*

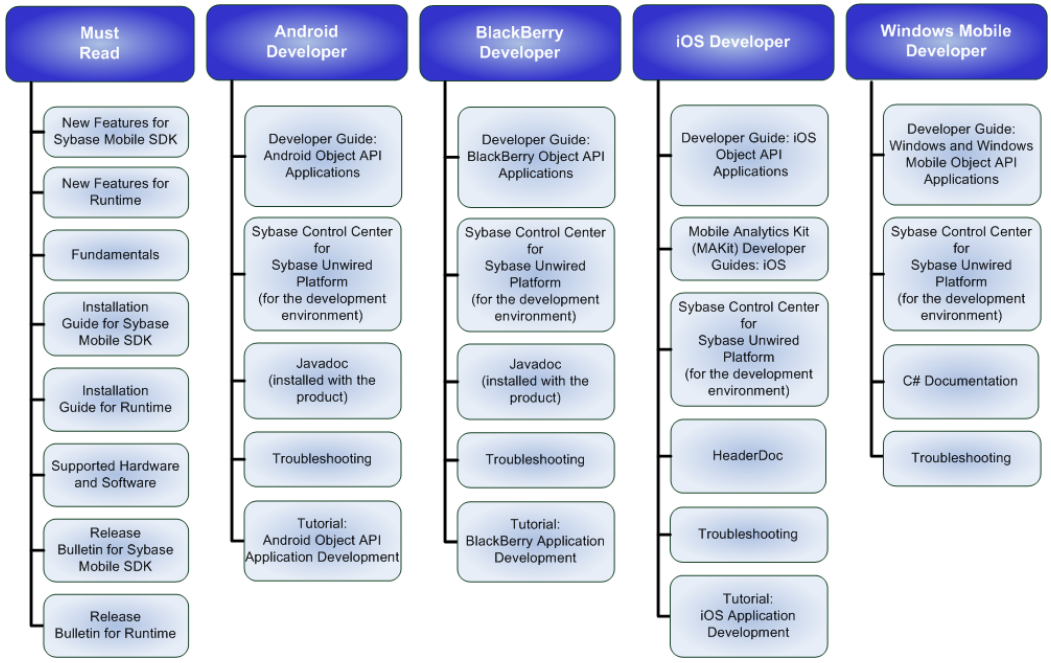
Mobile Data Models Development Documentation

These roles encompass development of mobile data models:

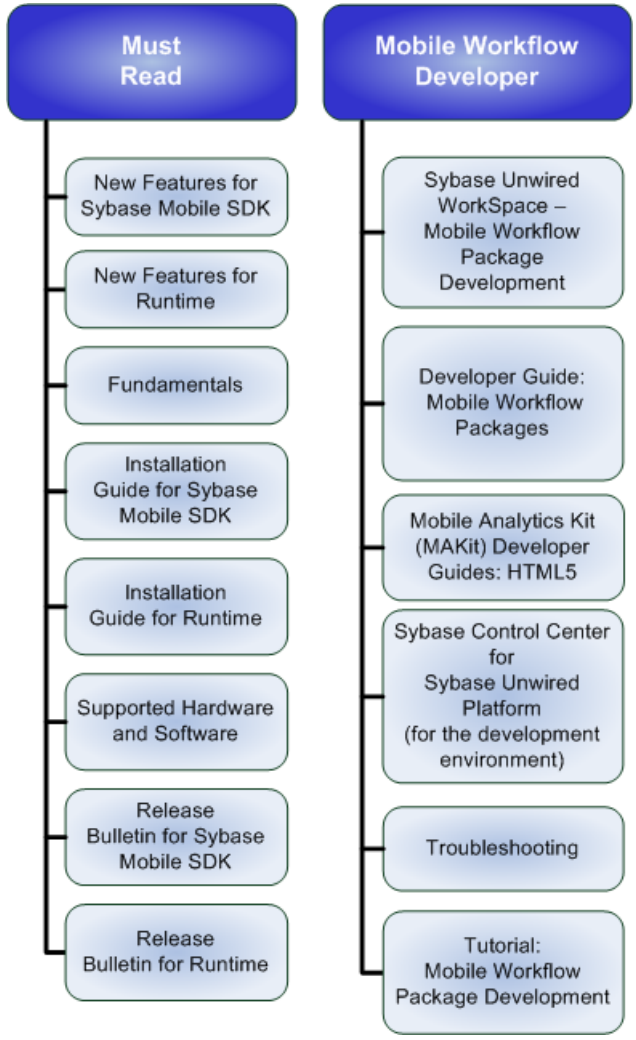
- **Mobility Solutions Architect –** Uses the tools to model the interaction between the mobile applications and existing EIS, services, and applications, or integrate mobile applications developed with other tools. Understands mobility concepts.
- **Mobile Business Objects (MBO) Developer –** Develops mobile business objects using Unwired WorkSpace, and the Unwired Server API. Generates native application code for BlackBerry, 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.


Must Read
New Features for Sybase Mobile SDK
New Features for Runtime
Fundamentals
Installation Guide for Sybase Mobile SDK
Installation Guide for Runtime
Supported Hardware and Software
Release Bulletin for Sybase Mobile SDK
Release Bulletin for Runtime
Mobility Solutions Architect
Sybase Unwired WorkSpace – Mobile Business Object Development
Sybase Unwired WorkSpace – Mobile Workflow Package Development
Sybase Control Center for Sybase Unwired Platform (for the development environment)
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
Sybase Unwired WorkSpace – Mobile Business Object Development
Sybase Control Center for Sybase Unwired Platform (for the development environment)
Troubleshooting
Tutorial: Mobile Business Object Development
Javadocs (installed with product)
DOE Developer
Mobile Data Models: Using Data Orchestration Engine
Sybase Control Center for Sybase Unwired Platform (for the development environment)
Troubleshooting
Javadocs (installed with product)
Gateway Developer
SAP NetWeaver Gateway documentation
Developer Guide: OData SDK (for API information)
Sybase Control Center for Online Data Proxy (for the development environment)
Troubleshooting
Javadocs (installed with product)

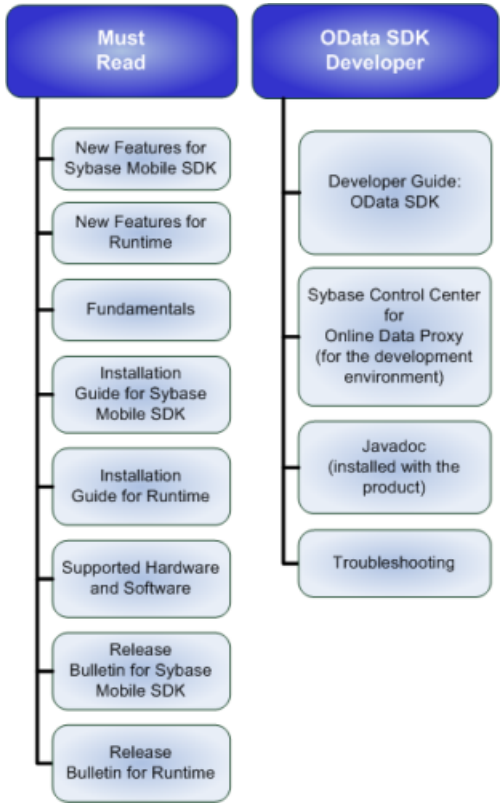
Object API Application Development Documentation



HTML5/JS Hybrid App Development Documentation



OData SDK Application Development



Documentation for Administrators

If you are an administrator, identify documentation based on your administrative focus.

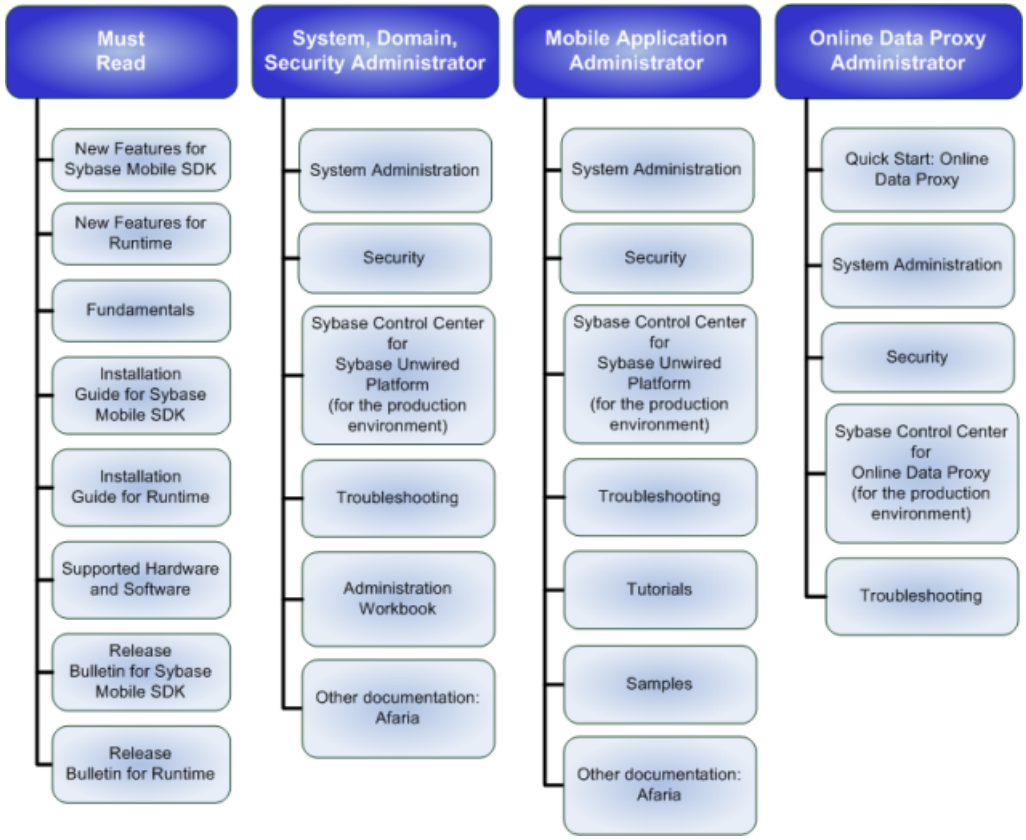
Convenient diagrams are provided to guide you through documentation for the following administration categories:

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

Runtime Administration Documentation

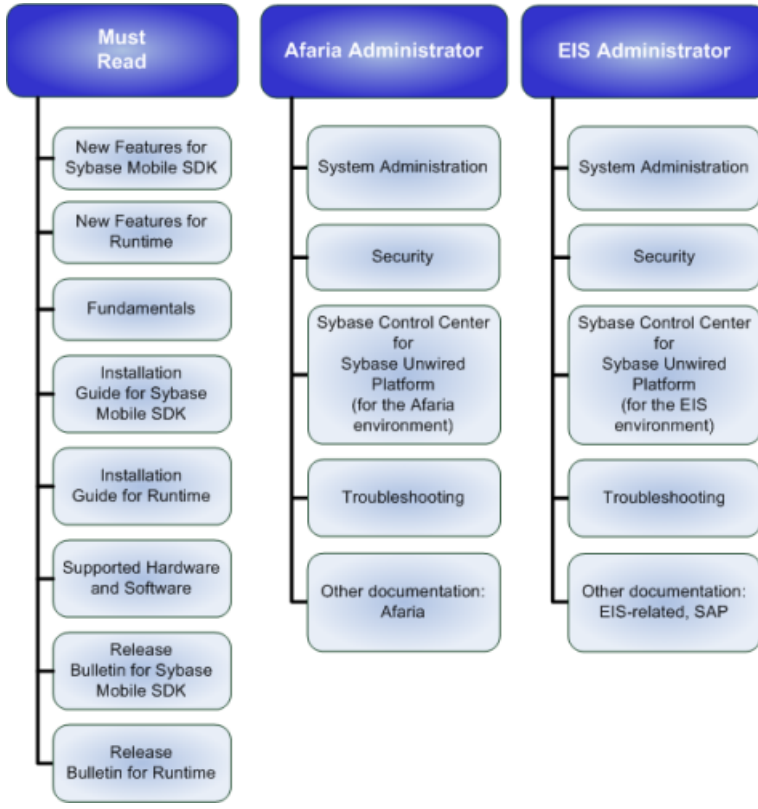
Runtime administration encompasses multiple roles:

- **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.
- **Online Data Proxy Administrator –** Administers the Online Data Proxy option.



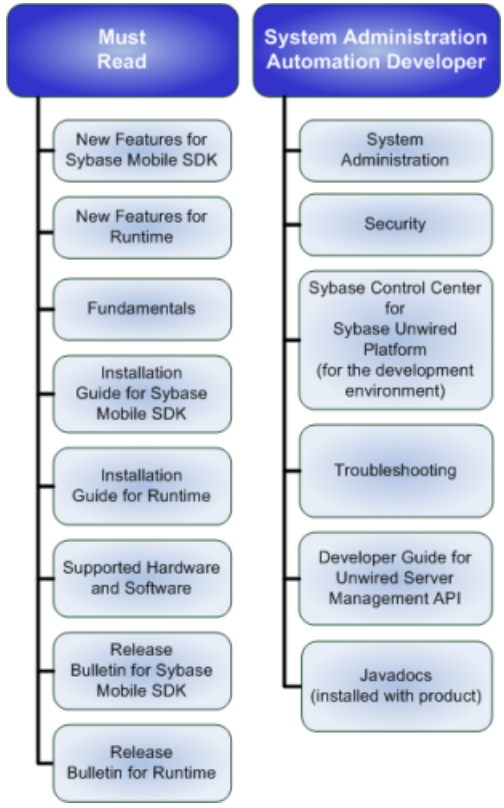
External Systems Administration Documentation

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



Administration Automation Development Documentation

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



Documentation Roadmap for Unwired Platform

Learn more about Sybase® Unwired Platform documentation.

Document	Description
New Features for Sybase Mobile SDK *New Features for Runtime*	Describes new or updated features. Audience: all users. Use: any time to learn what is available.
Fundamentals	Describes basic mobility concepts and how Sybase Unwired Platform enables you to design and implement mobility solutions. Audience: all users. Use: during the planning and installation phase, or any time for reference.

Document	Description
Troubleshooting	Provides information for troubleshooting, solving, or reporting problems. Audience: IT staff responsible for keeping Sybase Unwired Platform running, developers, and system administrators. Use: during installation and implementation, development and deployment, and ongoing maintenance.

Sybase Mobile SDK

Document	Description
Installation Guide for Sybase Mobile SDK	Describes how to install or upgrade Sybase Mobile SDK. Check the corresponding *Sybase Unwired Platform Release Bulletin* for additional information and corrections. Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system. Use: during the planning and installation phase.
Release Bulletin for Sybase Mobile SDK	Provides information about known issues, and updates. The document is updated periodically. Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information. Use: during the planning and installation phase, and throughout the product life cycle.

Document	Description

Tutorials	Tutorials for trying out basic development functionality.
	Audience: new developers, or any interested user.
	Use: after installation.
	• Learn mobile business object (MBO) basics, and use this as a foundation to create mobile device applications and mobile workflow applications:
	  • *Tutorial: Mobile Business Object Development*
	• Create native mobile device applications:
	  • *Tutorial: Android Object API Application Development*
	  • *Tutorial: BlackBerry Object API Application Development*
	  • *Tutorial: iOS Object API Application Development*
	  • *Tutorial: Windows Mobile Object API Application Development*
	• Create a mobile workflow package:
	  • *Tutorial: Mobile Workflow Package Development*
	Design process and best practices for developing MBOs.
	Audience: new and experienced developers.
	Use: after system installation.

Mobile Data Models: Using Data Orchestration Engine	Design process and best practices for developing mobile data models using Data Orchestration Engine. Audience: new and experienced developers. Use: after system installation.
Sybase Unwired WorkSpace – Mobile Business Object Development	Online help for developing MBOs. Audience: new and experienced developers. Use: after system installation.
Sybase Unwired WorkSpace – Mobile Workflow Package Development	Online help for developing mobile workflow applications. Audience: new and experienced developers. Use: after system installation.

Document	Description
Developer guides for native Object API and mobile workflow device application customization	Information for client-side custom coding using the Client Object API. Audience: experienced developers. Use: to custom code client-side applications. • *Developer Guide: Android Object API Applications* • *Developer Guide: BlackBerry Object API Applications* • *Developer Guide: iOS Object API Applications* • *Developer Guide: Windows and Windows Mobile Object API Applications* • *Developer Guide: Mobile Workflow Packages*
Developer Guide: OData SDK	Information for developing OData applications using the API. Audience: experienced developers. Use: to customize mobile applications using a proxy connection to an SAP OData source.
Mobile Analytics Kit (MAKit) Developer Guides: HTML5	Information for adding mobile analytics functionality to HTML5/JS Hybrid applications. Audience: experienced developers. Use: to add mobile analytics components for device applications. Dependencies: Use with *Developer Guide: Mobile Workflow Packages*.
Mobile Analytics Kit (MAKit) Developer Guides: iOS	Information for adding mobile analytics functionality to iOS Object API applications.. Audience: experienced developers. Use: to add mobile analytics components for device applications. Dependencies: Use with *Developer Guide: iOS Object API Applications*.

Runtime

Document	Description
Installation Guide for Runtime	Describes how to install or upgrade Sybase Unwired Platform Runtime . Check the corresponding *Sybase Unwired Platform Release Bulletin* for additional information and corrections. Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system. Use: during the planning and installation phase.
Release Bulletin for Runtime	Provides information about known issues, and updates. The document is updated periodically. Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information. Use: during the planning and installation phase, and throughout the product life cycle.
System Administration	Describes how to plan, configure, manage, and monitor Sybase Unwired Platform. Use with the *Sybase Control Center for Sybase Unwired Platform* online documentation. Audience: installation team, test team, system administrators responsible for managing and monitoring Sybase Unwired Platform, and for provisioning device clients. Use: during the installation phase, implementation phase, and for ongoing operation, maintenance, and administration of Sybase Unwired Platform.
Security	Describes how to plan, configure, and manage end-to-end security. Audience: installation team, test team, system administrators responsible for managing security. Use: during the installation phase, implementation phase, and for ongoing operation to secure the system, data, access, and authorization.

Document	Description
Quick Start: Online Data Proxy	Guides the administrator through the set of tasks to install, configure, manage, and monitor Online Data Proxy. Audience: system administrators responsible for managing proxy connections and Online Data Proxy mobile applications in the Sybase Unwired Platform runtime environment. Use: while getting started with Online OData Proxy
Sybase Control Center for Sybase Unwired Platform *Sybase Control Center for Online OData Proxy*	Describes how to use the Sybase Control Center administration console to configure, manage and monitor Sybase Unwired Platform. The online documentation is available when you launch the console (**Start > Programs > Sybase > Sybase Control Center**, and select the question mark symbol in the top right quadrant of the screen). Audience: system administrators responsible for managing and monitoring Sybase Unwired Platform, and system administrators responsible for provisioning device clients. Use: for ongoing operation, administration, and maintenance of the system.
Developer guide for system administration customization – *Developer Guide for Unwired Server Management API*	Information for custom coding using administration APIs. Audience: experienced developers. Use: to customize and automate administration at a coding level. Dependencies: Use with *Fundamentals* and *System Administration*.

Index

A

Afaria 33
    described 29
application deployment concepts 36
application life cycle 17
architecture 29
attributes 10

B

best practices for application development 21

C

change control 18, 21
clusters 34
code generation 12

D

data cache 11
data model 10
        See also MBO model
deploying MBOs 13
deployment packages 13
development cycle 18
development environment 17, 21
development life cycle 17
    testing 20
development task flow 18
documentation roadmap
    overview 41
DOE-based applications 14

G

generate client object code 12

I

intermediate release
    defined 18
    deployment package 18

K

keeping the data cache current 11

L

life cycle
    defined 17

M

maintenance cycle 21
MBO model
    defined 17
    new versions 21
mobility patterns 9

O

object code generation 12
object queries 11
operations 10

P

personalization keys 11
production cycle 20
production environment 17, 21

R

relationships 10
Relay Server 29
result set
    checkers 11
    filters 11

S

SAP Data Orchestration Engine (DOE) 14
security concepts 34
Sybase Control Center 29

Sybase Unwired Platform
development environment 17
production environment 17
test environment 17
synchronization parameters 11

T

test cycle 19
test environment 17, 21
testing 20
data handling 21
mobile applications on devices 21
mobile applications on simulators and emulators 21

U

Unwired Platform Runtime
architecture 29
Unwired Server
described 29
services provided by 31