# Developing Data Synchronization Solutions with webMethods Mobile Support

Version 9.7

October 2014

This document applies to webMethods Mobile Support Version 9.7 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.



**Document ID: IS-MS-DG-97-20141015**

# Table of Contents

# About this Guide

This guide describes how to use webMethods Mobile Support to synchronize data between mobile devices and a backend application.

To use this guide effectively, you should know how to create mobile applications using webMethods Mobile Designer. You should also know how to create webMethods Integration Server flow services.

## Document Conventions

| Convention | Description |
|---|---|
| **Bold** | Identifies elements on a screen. |
| Narrowfont | Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* . |
| UPPERCASE | Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+). |
| *Italic* | Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text. |
| `Monospace font` | Identifies text you must type or messages displayed by the system. |
| { } | Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols. |
| \| | Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the \| symbol. |
| [ ] | Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols. |
| ... | Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...). |

# Documentation Installation

You can download the product documentation using the Software AG Installer. The documentation is downloaded to a central directory named _documentation in the main installation directory (SoftwareAG by default).

# Online Information

### Software  AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products and certified samples, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center

### Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

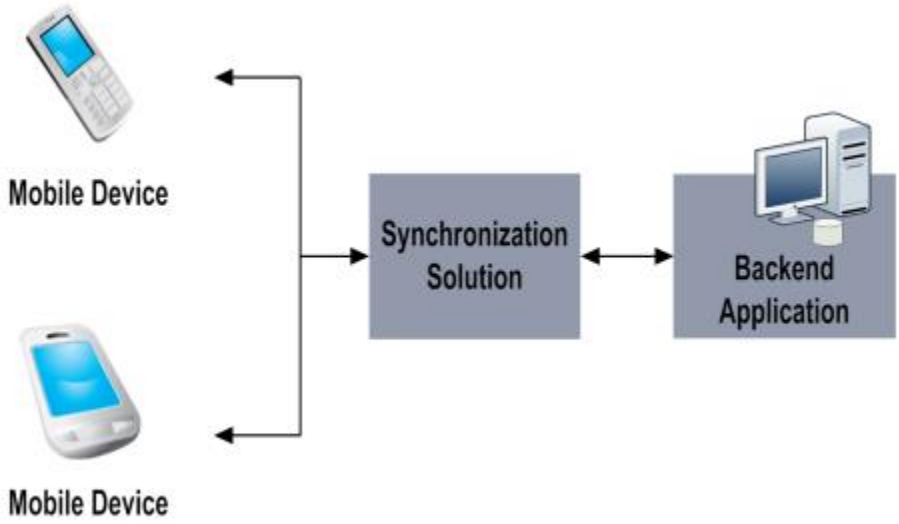
# 1 Overview

# What Is Mobile Data Synchronization?

Mobile devices have transformed our workplace and our personal and business interactions by providing the ability to have instant information at our fingertips. Almost every organization has "Mobile" in its strategic plans as a way to enable consumers, customers, and employees to share data. This trend has fueled the demand to expose, to mobile devices, business functionality that was normally reserved for "in-office" applications.

Mobile data synchronization solutions provide a way to accomplish the following:

- Transfer data back and forth between mobile devices and enterprise applications and their databases.
- Ensure data consistency between mobile devices and the enterprise application by resolving any data conflicts that result from multiple simultaneous updates.
- Enable the server infrastructure to be scalable.




Challenges exist that are unique to the mobile environment, such as unreliable networks, low bandwidth speeds, and devices that are connected only occasionally to a network. Mobile data synchronization solutions offer the following advantages:

- **Reduced dependency on network connections.** Employees can work offline on a mobile device and then synchronize their work when they are connected to the server, thereby eliminating the dependency on a steady network connection.
- **Improved data access speeds.** During synchronization, only the changed data is transferred between the mobile device's local database and the backend application. This approach alleviates network bandwidth limitations and improves data access speeds.

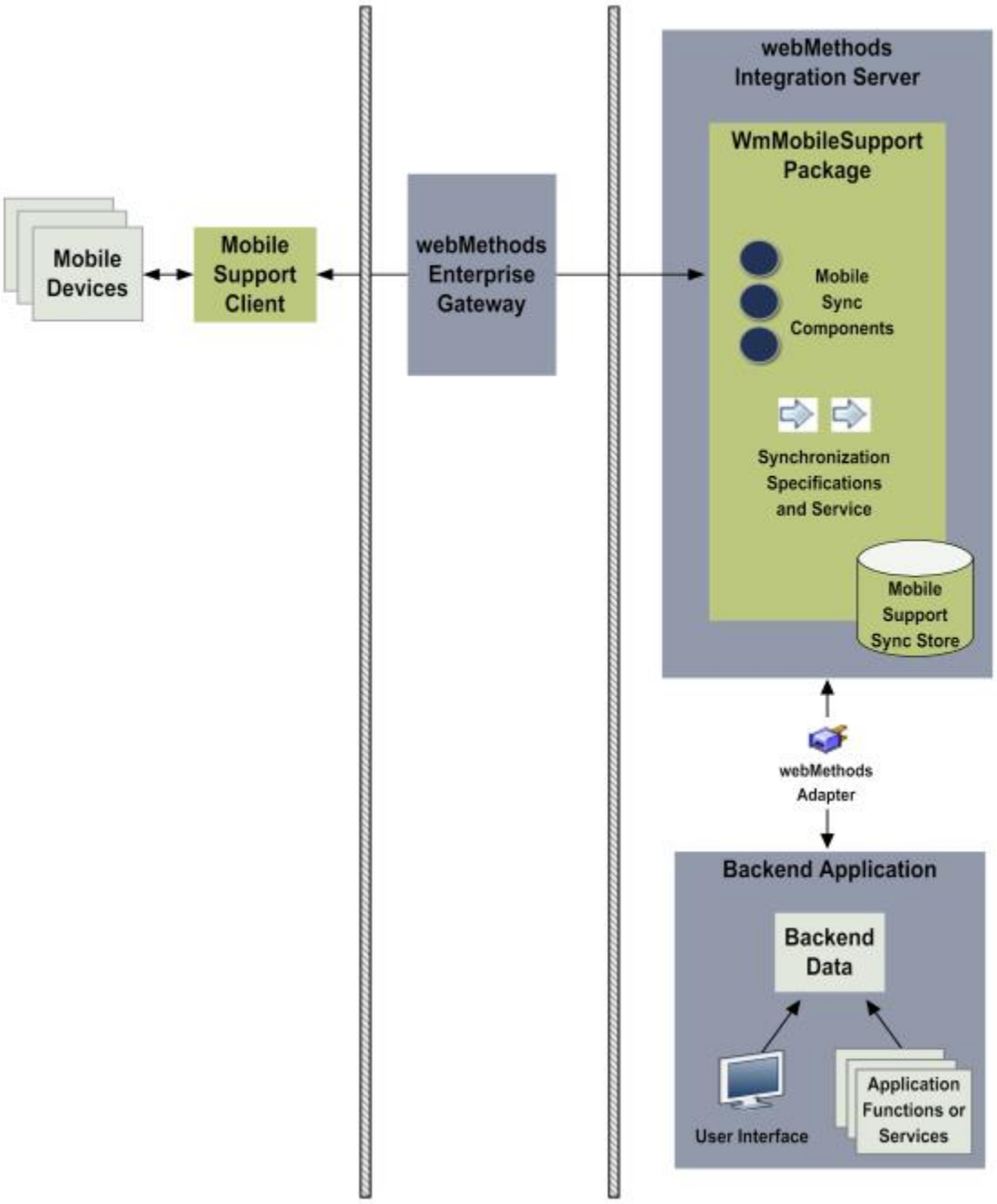
- **More accurate data.** Replacing paper data entry with the use of a mobile device helps improve the accuracy of the data. For example, the practice of capturing inventory data on paper, and manually transferring the data to an IT system later, is error prone and time consuming. Such a practice can lead to issues in order management and the supply chain. Entering this data on a mobile device, and then uploading the data when the device is connected to the server, helps automate the process and improve the accuracy of the data.

## What Is webMethods Mobile Support?

Software AG facilitates mobile data synchronization by way of webMethods Mobile Support, a feature of webMethods Enterprise Gateway. Mobile Support consists of the following client and server components:

- **Client component:** The Mobile Support Client provides an API library for use in mobile applications developed using webMethods Mobile Designer to initiate synchronization requests to a server. The Mobile Support Client, offered separately in the **Mobile** section of the Software AG Installer, should be installed on the system where mobile applications are developed.
- **Server component:** A webMethods Integration Server package called WmMobileSupport provides the tools the server needs to process synchronization requests received from a mobile device. In an Enterprise Gateway scenario, during installation of the internal Integration Server behind the firewall, the WmMobileSupport package is installed on the host internal server when **Mobile Support** is selected in the installer. For more information about the role of the internal Integration Server in an Enterprise Gateway configuration, see *webMethods Integration Server Administrator's Guide*.

The following diagram illustrates where the Mobile Support components fit in a scenario in which data is being synchronized between mobile devices and a backend application.




When a mobile device user submits a request to synchronize data with a backend application, the Mobile Support Client routes the request to Mobile Support on a host internal Integration Server behind the firewall. Mobile Support connects to the backend application using adapter logic. Mobile Support then executes services that pass data from the mobile device to the backend application, and from the backend application back to the mobile device, as defined for that particular mobile application.

In this scenario, “backend application” refers to an enterprise application that passes data using a mechanism such as a transfer file, a message on a queue, or a record in

a database. The data store associated with the backend application is considered the "master" set of data. Mobile Support updates this master set of data with data passed from a mobile application. The backend application can also write to this data store directly, outside of Mobile Support, by way of the application's user interface or its own set of functions or services.

The sections that follow provide a more detailed description of the role of each Mobile Support component.

## Mobile Support Client

The Mobile Support Client contains APIs that are used to initiate requests to Mobile Support for synchronizing data between a mobile device and a backend application. Mobile application developers invoke these methods in their mobile applications using webMethods Mobile Designer. For more information about using the Mobile Support Client's classes and methods, see "Step 3: Write the Mobile Application" on page 29.

## WmMobileSupport Package

On the server side, the WmMobileSupport package contains the following:

- **Built-in service and specifications.** Using Software AG Designer with the Service Development perspective, business integration developers write flow services that include Mobile Support's built-in service and specifications. The flow services pass data to and from the backend application and notify Mobile Support when the backend application has made a change to the backend data directly, outside of Mobile Support.

  For more information about using these specifications and the service, see "Step 1: Write the Server-Side Business Logic" on page 20.

- **Mobile sync components.** Using webMethods Integration Server Administrator, business integration developers create mobile sync components to supply information that the server needs to process synchronization requests received from a mobile device.

  For more information about mobile sync components, see "Mobile Sync Components" on page 12.

## Mobile Support Sync Store

Mobile Support uses a database that resides on Integration Server, referred to as the Mobile Support*sync store*, to record the latest changes that were made to the backend data by either the mobile application or the backend application. In addition to recording the type of change that was made to a data record (create, update, or delete), Mobile Support uses the sync store to detect and resolve data conflicts. For more information about data conflicts, see "Change Detection and Conflict Resolution" on page 13.

## Mobile Sync Components

Mobile sync components specify the details that Mobile Support needs to process synchronization requests received from a mobile device. These details include the following:

- Download and upload services to invoke to handle the data being synchronized
- Structure of the data being synchronized
- Row identifiers that uniquely identify the data record being synchronized
- Optional filters to specify a subset of data to return to the mobile application (for example, if a mobile inventory application user requests to see only the products in a particular category)
- Rule to apply if a data conflict arises
- Whether to store a copy of the data being synchronized, referred to as the *business data*, to assist with detecting data changes when the backend application updates the backend data directly

By default, Mobile Support does not store business data in the sync store. When changes are made to the backend data only by mobile applications that submit synchronization requests to Mobile Support, storing business data in the sync store is unnecessary and can result in additional overhead. However, when the backend data will also be updated by the backend application itself, Software AG recommends configuring the mobile sync components to store business data in the sync store to assist Mobile Support in detecting and resolving data conflicts. For instructions on configuring this option, see "Step 2: Create a Mobile Sync Component" on page 26.

For more information about creating and managing mobile sync components, see "Developing Solutions Using Mobile Support " on page 19.

## webMethods Adapter

Mobile Support requires a mechanism to connect with the backend application, to update the backend data with data that Mobile Support receives from a mobile device, and to notify the server and refresh the sync store entries when the backend application updates the backend data directly. The procedures in this guide assume that you are using a webMethods Adapter for this purpose.

## Data Synchronization in a Clustered Environment

When Mobile Support is used in a clustered environment with a load balancer, the load balancer distributes data synchronization requests among the host Integration Servers for processing. To ensure that all servers in the cluster are working with the same set of data, Integration Server uses the Terracotta Server Array to facilitate synchronization.

For more information, see "Considerations for Using Mobile Support in a Clustered Environment" on page 17.

## Change Detection and Conflict Resolution

When a mobile application user downloads data from the server, the data the server sends to the device is a copy of all or part of the backend application's "master" data. Conflicts can arise when the backend master data is updated by multiple sources. Examples are as follows:

- Conflicts can arise if multiple mobile application users submit changes to the same backend data around the same time. When changes that a user submits are no longer based on current master data, the submitted data is referred to as *stale data*.
- Conflicts can arise if the backend application updates the backend data directly and a mobile application user has submitted changes to the same backend data.

When Mobile Support receives a synchronization request, the server must determine whether a conflict exists between the data the mobile application user sends and the backend application's version of the data. If the backend data is updated solely through requests received from mobile devices through Mobile Support, Mobile Support will ensure that no conflicts exist between devices. If the backend data will also be updated by the backend application, Mobile Support will also consider the backend application's updates when looking for conflicts.

If a conflict exists, Mobile Support resolves the conflict using the *conflict resolution rule* that the business application developer specifies during creation of the mobile sync component. Mobile Support offers the following conflict resolution rules:

- **"Client wins" rule (accept stale updates):** With this conflict resolution rule, Mobile Support accepts the data the mobile application sends, even if the corresponding backend record was changed since the device last synchronized with the server.
- **"Server wins" rule (reject stale updates):** With this conflict resolution rule, Mobile Support ignores the data the mobile application sends if the corresponding backend record was changed since the device last synchronized with the server.

  For this rule, if the device's data is stale, Mobile Support overwrites the device's data with the latest data from the server. Mobile Support then returns a response to the calling application indicating that the conflict was resolved by using the server's version of the data.

  If the device's data is more current than the server's data, Mobile Support overwrites the backend data with the data from the mobile device.

# 2 Configuring webMethods Mobile Support

# Before You Configure Mobile Support

Before you start performing the configuration tasks described in this topic, make sure the following components are in place:

- Enterprise Gateway license and at least one internal Integration Server, installed behind the firewall as described in *Installing webMethods and Intelligent Business Operations Products*
- Mobile Support, installed on the internal Integration Server
- Software AG Designer with the Service Development perspective
- Mobile Support database component called MobileSupport, created using the webMethods Database Component Configurator as described in *Installing webMethods and Intelligent Business Operations Products*
- Connection configured to the database that Mobile Support will use as the sync store
- A webMethods adapter or similar mechanism to implement adapter logic, installed on the Integration Server that hosts Mobile Support
- Mobile Designer and the Mobile Support Client, installed on the system where mobile applications are developed

# Configuring Mobile Support

Perform the following steps to set up the Mobile Support environment for mobile data synchronization.

### To configure Mobile Support

1. Open the Integration Server Administrator if it is not already open.
2. On the Settings > JDBC Pools screen, ensure that the MobileSupport functional alias is configured to point to the Mobile Support sync store. The JDBC connection information was defined during installation. If you need to make changes to the settings, keep the following points in mind:
   - All mobile sync components that are enabled should be disabled before changes are made to the database connection URL. You can enable the mobile sync components after you save your changes and restart Integration Server. For details, see "Managing Mobile Sync Components" on page 37.
   - Changing the database connection URL will result in the loss of all data in the sync stores associated with any mobile sync components that have been created.
   - Changes should be made using the Integration Server Administrator. Changing JDBC pool settings in a text editor can introduce errors.

For more information about configuring functional aliases, see *Installing webMethods and Intelligent Business Operations Products*.

## Considerations for Using Mobile Support in a Clustered Environment

If you are using Mobile Support in a clustered Integration Server environment, keep the following points in mind:

- The MobileSupport functional alias must point to the same Pool Definition values in all the nodes of the cluster.
- Configuration of all mobile sync components must be identical across all nodes of the cluster. As part of this configuration, all packages containing the business document types and download and upload services specified in each mobile sync component must exist on each node of the cluster. You can use webMethods Deployer to help automate this process.
- Mobile Support uses the locking and unlocking functionality of Terracotta Ehcache to achieve synchronization. In a clustered environment, Mobile Support also uses a system cache, maintained on the Terracotta Server Array, to enable all Integration Servers in the cluster to process data synchronization requests received from mobile applications. Software AG recommends that you not edit the configuration parameters related to this system cache.

For more information about the various tasks involved with configuring and using Mobile Support in a clustered environment, see the following resources.

| For more information about... | See... |
|---|---|
| Configuring Integration Server clusters | *webMethods Integration Server Clustering Guide* |
| Configuring functional aliases | *Installing webMethods and Intelligent Business Operations Products* |
| Replicating Integration Server packages | *webMethods Integration Server Administrator's Guide* |
| Using webMethods Deployer to automate mobile sync component configuration | "Deploying and Migrating Mobile Support Configuration" on page 18 |
| How Mobile Support uses Terracotta and the Mobile Support | *Getting Started with the webMethods Product Suite and Terracotta* |

| For more information about... | See... |
| --- | --- |
| system cache in a clustered environment | |

## Deploying and Migrating Mobile Support Configuration

Mobile Support maintains configuration information for mobile sync components and mobile applications in the following files in the *Integration Server_directory*/instances/*instance_name* /packages/WmMobileSupport/config directory:

- **mobileSyncComponents.cnf.** This file contains the following information:
  - Mobile sync component alias
  - Names of the download and upload service that will handle the data being synchronized
  - Name of the business document type that defines the structure of the data being synchronized
  - Conflict resolution rule
  - Row identifiers
  - Filter definitions, if provided, that specify a subset of data to return to the mobile application
  - Whether the mobile sync component is configured to store business data in the sync store
  - State of the mobile sync component (enabled, disabled, suspended)
- **mobileApp.cnf.** This file contains the name and version of the mobile application and the mobile sync components with which the application is associated.

These configuration files are assets that can be deployed and migrated.

When you use webMethods Deployer to deploy these configuration files to another Integration Server, deploy both files together. Also, ensure that the business document type and the download and upload services specified for the mobile sync components are present on the target server. For details about deploying assets using Deployer, see *webMethods Deployer User's Guide*.

When you upgrade the host Integration Server to a new release, the migration utility locates and migrates these configuration files automatically when you respond "yes" to the prompt asking if you want to migrate Mobile Support data. During the upgrade, be sure to migrate the packages that contain the business document type and the download and upload services specified for the mobile sync components configured in the mobileSyncComponents.cnf file. For details about upgrading Integration Server, see *Upgrading webMethods and Intelligent Business Operations Products*.

# 3 Developing Solutions Using Mobile Support

# Overview

Developing mobile data synchronization solutions using Mobile Support consists of the following high-level steps:

1. The business integration developer writes the business logic on the server side using Software AG Designer with the Service Development perspective. This business logic consists of an IS document type, as well as flow services and adapter services that send data received from a mobile device to the backend application, return changed data from the backend application to the mobile device, and notify Mobile Support when the backend data is updated directly by the backend application.
2. The business integration developer creates a mobile sync component, using Integration Server Administrator, to supply information that Mobile Support needs to process synchronization requests received from a mobile device.
3. The mobile application developer writes a mobile application, using webMethods Mobile Designer and the Mobile Support Client, to initiate requests to Mobile Support for synchronizing data between a mobile device and a backend application.
4. The business integration developer associates the mobile application with a mobile sync component using Integration Server Administrator.

# Step 1: Write the Server-Side Business Logic

**Who performs:** Business integration developer

**Development tools needed:**Software AG Designer with the Service Development perspective

The server-side business logic consists of the following:

- **An IS document type.** This document type, referred to as the *business document type*, defines the structure of the data you are synchronizing.
- **Adapter services.** These services connect to the backend application and retrieve the data to synchronize.
- **Download and upload flow services.** These services function as wrappers for the adapter services to pass data between the mobile device and the backend application. They contain the logic needed to identify the data to access and any transformation or mapping that may be needed for that data.
- **Notification services.** These services alert Mobile Support when the backend application has modified the backend data directly, and then update the Mobile Support sync store to match the changes made to the backend data. Notification services require the creation of adapter notifications and IS triggers.

Before you start developing the business logic, identify the structure of the data to be synchronized.

## Creating the Business Document Type

Before you start writing the adapter services and the upload and download flow services, create the business document type that represents the structure of the data being synchronized. When creating the business document type, keep the following points in mind:

- Supported data types for the business document type are as follows:
  - String
  - Document
  - Document Reference
  - Object, declared as one of the following basic Java classes that Integration Server and Mobile Support support natively: boolean, byte, character, date, double, float, integer, long, or short
- Include one or more fields that can uniquely identify each record being synchronized. The value of these fields should not change. Examples include a sequence number in a primary key or a user ID.

You will supply the business document type and the unique identifier fields when you create a mobile sync component. For more information, see "Step 2: Create a Mobile Sync Component" on page 26.

For more information about creating document types, see *webMethods Service Development Help*.

## Writing the Adapter Services

Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from Integration Server. In a mobile data synchronization solution, adapter services specify the backend application fields that correspond to the fields specified in the business document type and the connection information that Mobile Support will use to connect to the backend application's data store.

The adapter services that you create for a data synchronization solution depend on the type of adapter you are using and the type of flow service that will invoke the adapter services.

| For this type of flow service... | Create the following... |
| --- | --- |
| Download flow service | ■ An adapter service that retrieves all of the backend data when the *fetchAll* parameter in the |

| For this type of flow service... | Create the following... |
| --- | --- |
| | wm.mobile.datasync.specs:downloadSpec specification is set to `Yes`<br>■ An adapter service that retrieves only a specific record when the *fetchAll* parameter in the specification is set to `No` |
| Upload flow service | ■ An adapter service that handles update requests received from a mobile device<br>■ An adapter service that handles insert requests received from a mobile device<br>■ An adapter service that handles delete requests received from a mobile device |

For all adapter services written for mobile data synchronization, it is recommended to specify a result field to determine the success or failure of processing a particular record. This is particularly helpful for setting the *rowStatus* variable in the wm.mobile.datasync.specs:uploadSpec specification to 0 (to indicate success) or 1 (to indicate failure).

For more information about writing adapter services, see the documentation provided for the adapter you are using.

## Writing the Flow Services

You create the necessary flow services for uploading and downloading data to and from the backend application using Service Development.

For details about using Service Development to write flow services, see *webMethods Service Development Help*.

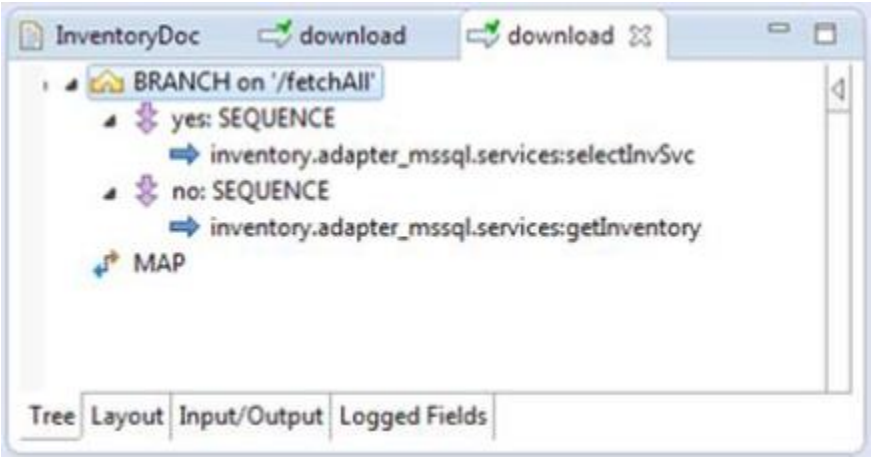
### Writing the Download Service

When writing the download service to return data from the backend application to a mobile device, include the following:

- Include a reference to the wm.mobile.datasync.specs:downloadSpec specification. This specification defines the input and output signatures of the download service.
- Include branching logic to handle the specified amount of backend data to fetch. In the branch step, call the appropriate adapter service to download all data or only certain records according to the value specified in the *inputArgs* and *fetchAll* parameters of the wm.mobile.datasync.specs:downloadSpec specification.

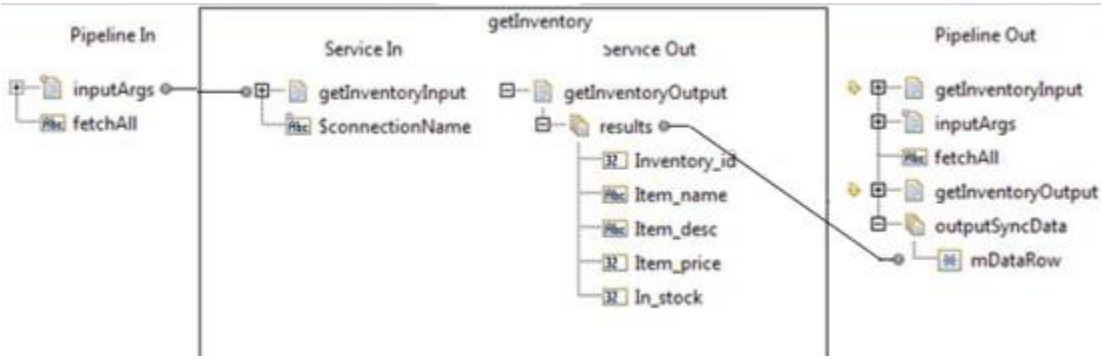
  In the following example, the download service invokes an adapter service called selectInvSvc to select all inventory records from the backend application when *fetchAll*

is set to "yes." When *fetchAll* is set to "no," the download service invokes an adapter service called getInventory to select a single inventory record from the backend application.




- In the pipeline, map the *inputArgs* input parameter from the wm.mobile.datasync.specs:downloadSpec specification to the adapter service's input signature when *fetchAll* is "no." Then, map the variables from the fetched data in the adapter service's output to the corresponding variables in the specification's *outputSyncData* output parameter.

  In the example described above, the selectInvSvc service does not require input arguments because the service is defined to fetch all inventory records. The getInventory service requires input arguments to specify which record to fetch. In this case, the mapping might look like the following:



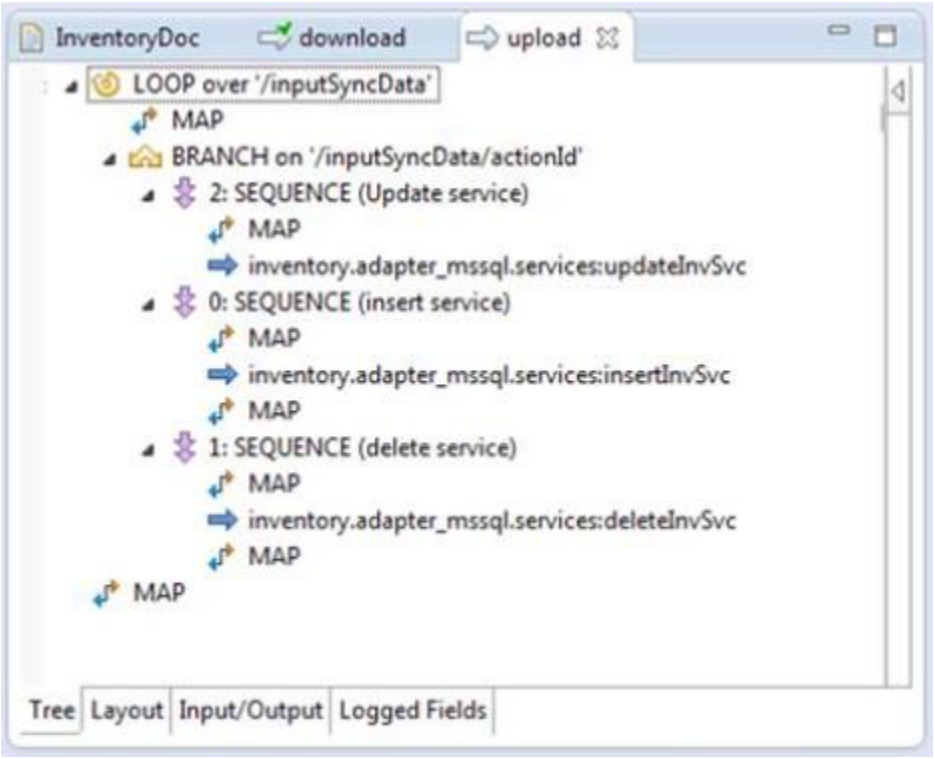

  In the pipeline input, *inputArgs* is a document that contains the fields specified in the mobile sync component's row identifier parameter.

  **Note:** The document mapped to *mDataRow* should match the business document type you created in "Creating the Business Document Type" on page 21.

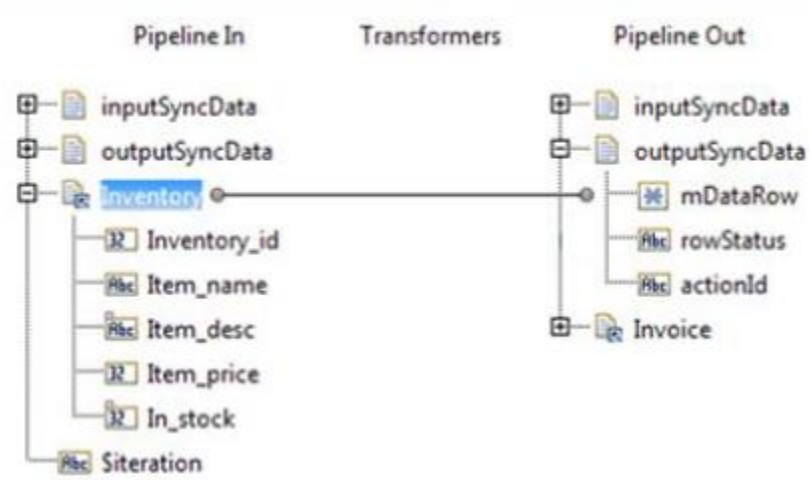
## Writing the Upload Service

When writing the upload service to pass changed data from the mobile device to the backend application, include the following:

- Include a reference to the wm.mobile.datasync.specs:uploadSpec specification. This specification defines the input and output signatures of the upload service.
- Include a LOOP step to process each record received from the mobile application.
- Within the LOOP step, include branching logic to call the appropriate adapter service to insert, delete, or update data received from the mobile device based on the service's *actionId* input parameter. An action ID of `0` indicates to insert the record, `1` indicates to delete the record, and `2` indicates to update the record. For example:




- For each iteration of the LOOP step, *outputSyncData* should contain an *mDataRow* entry for data being inserted or updated. Map the inserted or updated record to the *mDataRow* variable beneath the *outputSyncData* document in the pipeline's output. For example:




**Note:** The document mapped to *mDataRow* should match the business document type you created in "Creating the Business Document Type" on page 21.

## Writing the Notification Services

Notification services alert Mobile Support when the backend application updates the backend data directly, outside of Mobile Support.

**Note:** When the backend data is updated by both the backend application and a mobile application that submits synchronization requests to Mobile Support, Software AG recommends configuring mobile sync components created for your synchronization solution to store business data in the sync store. Doing so helps ensure that Mobile Support can correctly detect and resolve conflicts. For instructions, see "Step 2: Create a Mobile Sync Component" on page 26.

Mobile Support determines whether a synchronization request comes from a mobile application or from a backend notification service based on the set of parameters provided in the synchronization request message. When the request header contains a mobile application name and a mobile application version, this indicates that the request is from a mobile application. If this information is not present, Mobile Support identifies the request as coming from a notification service.

**To write a notification service**

1. Create an adapter notification to contain information about an event that occurs on an adapter resource, and to send the notification data to Integration Server in the form of a published document.

   Configure notifications so that they are received in the order in which data changes occur. If you are using webMethods Adapter for JDBC, for example, you can specify an “ordered notification” type for this purpose.

**Note:** If you are using a polling adapter, keep in mind when setting the polling interval that the potential exists for a subsequent backend data update to occur before Mobile Support receives the notification about a previous update. Set the interval based on the expected frequency of backend data updates made directly by the backend application and the conflict resolution rule in place for the mobile application.

2. Create the trigger service that processes the published document. In a mobile data synchronization scenario, the published document contains the changed backend data that Mobile Support should use to update the sync store. When creating this service, keep the following points in mind:

   - Include branching logic to identify the operation executed (update, insert, or delete).
   - Invoke the wm.mobile.datasync:synchronize service to update the Mobile Support sync store with the changes made in the backend data. Pass the input payload as defined in the service's input signature.
   - Create a messaging trigger to subscribe to the document type created for the notification and invoke a trigger service to process a published document when one is received.

   For more information about creating adapter notifications, messaging triggers, and trigger services, see *webMethods Service Development Help* and the adapter documentation for the adapter you are using.

## Step 2: Create a Mobile Sync Component

**Who performs:** Business integration developer

**Development tools needed:**webMethods Integration Server Administrator

You create a mobile sync component to supply information that Mobile Support needs to process synchronization requests received from a mobile application.

Some applications use multiple datasets. For example, an inventory application may have one dataset for order entry and another dataset where product details are maintained. If your application uses more than one dataset, create one mobile sync component for each dataset, associate each mobile sync component with the appropriate mobile application, and ensure the mobile application includes the mobile sync component alias when it sends a request so that Mobile Support can route the request to the correct service.

Before you create a mobile sync component, make sure you have created the following items:

- Flow service for uploading data from the mobile device to the backend application
- Flow service for downloading data from the backend application to the mobile device

- Business document type that defines the structure of the data to be synchronized

For details about creating these items, see "Step 1: Write the Server-Side Business Logic" on page 20.

**To create a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. Click **Add Mobile Sync Component**.
5. On the Add Mobile Sync Component screen, enter the following information:

| For this parameter... | Specify... |
|---|---|
| **Alias** | A unique name for the mobile sync component. |
| **Download Service** | The fully qualified name of the flow service created to return data from the backend application to the mobile device. |
| **Upload Service** | The fully qualified name of the flow service created to send data from the mobile device to the backend application. |
| **Business Doc Type** | The fully qualified name of the IS document type to use to define the structure of the data to be synchronized. This document type was created earlier as described in "Creating the Business Document Type" on page 21.<br><br>Mobile Support uses this document type to construct the *mDataRow* variable in the wm.mobile.datasync.specs:downloadSpec and wm.mobile.datasync.specs:uploadSpec specifications implemented in the wm.mobile.datasync:synchronize service. |
| **Row Identifiers** | One or more field names from the business document type that you want to use to identify the record being synchronized. Separate field names with commas.<br><br>Mobile Support uses these fields to construct a single unique identifier for each record. |

| For this parameter... | Specify... |
| --- | --- |
| **Filter** | Optional. One or more keys that specify which records to return to the mobile application. Separate key names with commas.<br><br>The filter keys you specify here must be present in the business document type, and the mobile application should send these same keys in the synchronization request.<br><br>Mobile Support sends the requesting mobile application only the set of records that matches the filter, as follows:<br><br>■ If more than one filter key is passed, Mobile Support returns only those records that match all of the specified filter keys.<br><br>■ If a filter key contains more than one value, Mobile Support returns records that match any of the values. |
| **Conflict Resolution Rule** | The rule to use if a conflict occurs. Specify one of the following:<br><br>■ **Client wins:** Accept data the mobile application sends, even if the corresponding backend record was changed since the device last synchronized with the server. Specify this rule when your business case permits stale data.<br><br>■ **Server wins:** Ignore the data the mobile application sends if the corresponding backend record was changed since the device last synchronized with the server. Specify this rule when updates should only be made to current data.<br><br>For more information about conflict resolution rules, see "Change Detection and Conflict Resolution" on page 13. |
| **Store Business Data** | Whether to store, in the sync store, the business data to be synchronized. By default, Mobile Support does not store business data in the sync store.<br><br>**Important:** If the backend data is updated by the backend application, outside of Mobile Support, as well as by mobile applications that submit synchronization requests to Mobile Support, select this check box. Doing |

| For this parameter... | Specify... |
| --- | --- |
| | so assists Mobile Support in detecting and resolving data conflicts. |

6. Click **Save Changes**.

   By default, mobile sync components are disabled when they are first created. You will enable this mobile sync component in a later step.

## Step 3: Write the Mobile Application

**Who performs:** Mobile application developer

**Development tools needed:**webMethods Mobile Designer and the Mobile Support Client

Mobile applications that synchronize data between a mobile device and a backend application must invoke methods from the following classes in the Mobile Support Client library:

- com.softwareag.mobile.data.client.Context
- com.softwareag.mobile.data.sync.DataSynchronization
- com.softwareag.mobile.data.sync.Filter
- com.softwareag.mobile.data.sync.ResponseSet
- com.softwareag.mobile.data.sync.RowStructure

Mobile Designer copies these classes to the _temp_/_supportclient_ and _temp_/_src folders in your project when you activate a device. Mobile Designer obtains the location of these classes from the mobilesupportclient.runtime.dir property for the project. For details about setting this property, see "Preparing to Use the Mobile Support Client" on page 30.

Your mobile application should include functionality to create a Context instance and set initialization parameters. Depending on business need, your application should also include the logic to download data from the backend application, synchronize data with the backend application, and process the response. In addition, the application should contain logic to read data from, and write data to, the device's local database. For details on including this logic, see the sections that follow.

For more information about creating mobile applications using Mobile Designer, including creating a project, setting properties and parameters for the project, defining the devices you want your application to support, and distributing the application on the appropriate platforms, see *Using webMethods Mobile Designer*. For details about the Mobile Support Client classes, see *webMethods Mobile Support Client Java API Reference*.

## Preparing to Use the Mobile Support Client

Before you can use the Mobile Support Client library in your mobile application, you must specify where the Mobile Support Client is installed. You must also reference, in the project you create for your mobile application, the JSON API that the Mobile Support Client needs for device activation.

**To prepare to use the Mobile Support Client**

1. Open Mobile Designer if it is not already open.
2. Set the mobilesupportclient.runtime.dir property in the project's _default_.xml file to the path where the Mobile Support Client is installed. The default location is *Software AG_directory*\MobileDesigner\MobileSupportClient.

   For more information about this property, see *Using webMethods Mobile Designer*.
3. The Mobile Support Client uses the JSON API in Mobile Designer's "Library JSON" sample project. This project is available in the *Mobile Designer_directory*\Samples directory where Mobile Designer is installed. Reference this project as a built library in your Mobile Support Client project before performing device activation.

   For details about building and referencing libraries, see the section on creating and using code libraries in *Using webMethods Mobile Designer*.

## Creating a Context Instance

Your mobile application should include a Context instance. Before you write the logic to initiate synchronization requests, create the Context instance by calling the getContext method in the com.softwareag.mobile.data.client.Context class.

## Setting Initialization Parameters

Invoke the setInitializeConf method in the com.softwareag.mobile.data.client.Context class to pass the following initialization parameters to the Mobile Support Client:

- Host and port of the Enterprise Gateway Server through which the request will be routed
- Name and version of the mobile application
- Type of mobile device that is running the application
- User name and password used to authenticate the mobile application user with Integration Server
- Protocol to use to communicate with the server (`0`, the default, indicates to use the HTTP protocol, and `1` indicates to use the HTTPS protocol)

For more information about using the HTTPS protocol to connect to a server from a mobile application, see the section on installing custom SSL certificates on devices in *Using webMethods Mobile Designer*.

When a mobile application user submits a download or synchronization request, the Mobile Support Client invokes the wm.mobile.datasync:synchronize service on the server using the following URL:

*protocol* ://*host* :*port* /invoke/wm.mobile.datasync:synchronize

Add desired logic to handle SyncClientException exceptions that the setInitializeConf method returns when it encounters validation errors.

**Note:** If Enterprise Gateway mobile application protection filters are being used on the server to control access for certain mobile application versions on a predefined set of mobile platforms, make sure that the device type and application name and version you pass in the setInitializeConf method match those specified in the mobile application protection filter. For more information about mobile application protection filters in Enterprise Gateway rules, see *webMethods Integration Server Administrator's Guide*.

## Writing the Logic to Initiate Download Requests

Your mobile application should include logic to download backend data from the server to the mobile device. This logic is needed to accomplish the following:

- Populate the mobile device's local database with data when the mobile application receives a response for the download request.
- Overwrite the mobile device's local data with backend data from the server when needed in the following situations:
  - When you specify or change filters to download a subset of data
  - When you receive a "success" status from the server (status code `40`) indicating that the download request was processed successfully
  - When the device's data becomes corrupted

Include the following:

- If you want to define a filter that specifies a subset of data to download from the backend application (for example, to reduce the amount of data synchronized over a cellular network or to restrict downloaded data to a particular project), invoke the add method from the com.softwareag.mobile.data.sync.Filter class. Keep the following points in mind:
  - Invoke the add method once for each filter key that the business integration developer specifies in the corresponding mobile sync component.
  - If a particular filter key is defined in the mobile sync component and you do not want that key to be applied, set the value for the key to null.

- If you pass more than one filter key, Mobile Support returns a record to the mobile application only if the record matches all of the specified filter keys.
- You can specify multiple values for a filter key. If you do so, Mobile Support returns records that match any of the values.
- If you change a filter's key or values later, you must invoke the download method from the com.softwareag.mobile.data.sync.DataSynchronization class, as described later in this procedure, to download data that meets the revised filter criteria.
- You can invoke the getKeys method in the com.softwareag.mobile.data.sync.Filter class to see a list of filter keys you previously added. To see the values that are set for a specified filter key, invoke the get method. To see the number of filter key/value pairs defined, invoke the size method.

- Invoke the download method from the com.softwareag.mobile.data.sync.DataSynchronization class. This method passes the following parameters:
  - **Mobile sync component alias.** Obtain this alias from the business integration developer who created the mobile sync component.
  - **Filter.** If you used the add method as described previously to define a filter, pass the filter instance to the download method.

    If you did not define a filter using the add method, set the *filter* instance to null.
- The com.softwareag.mobile.data.sync.ResponseSet class encapsulates the response from the synchronization request. The response includes records from the server (filtered, if a filter was supplied with the request). Use the getRowStructures and getStatus methods in this class to process the response further.

  Mobile Support returns the following status codes:

| Status Code | Meaning |
|---|---|
| 30 | An error occurred in the configuration or state of the mobile sync component associated with the mobile application.<br><br>To resolve this error, work with the server administrator to ensure that the association between the mobile application version and the mobile sync component is valid and that the mobile sync component is enabled. If appropriate for your business need, have the mobile application provide the user with feedback about the error and instructions for resubmitting the data. |
| 31 | The mobile application passed a filter with the request, and a filter validation error occurred.<br><br>To resolve this error, ensure that the mobile application passes the expected number of filter keys and that the |

| Status Code | Meaning |
|---|---|
| | filter key names match those specified in the mobile sync component's filter definition. |
| 40 | Mobile Support processed the synchronization request successfully.<br><br>Clear the data corresponding to the mobile sync component from the mobile device's local database. Then, process the response set to write the new data received from the server to the local database. Including this logic in the application helps prevent data inconsistencies by providing the application with a fresh set of records.<br><br>**Note:** Clear and refresh the mobile device's local database only when the server returns a status code of 40. |
| 50 | Mobile Support encountered an internal error while processing the synchronization request.<br><br>Work with the server administrator to resolve this error. If appropriate for your business need, have the mobile application provide the user with feedback about the error and instructions for resubmitting the data. |

- Include logic to handle user interface messages based on the status messages received from Mobile Support.

  **Important:** If a mobile application user changes data on the device and then performs a "download" action (instead of a "sync" action), the user's local changes will be lost when Mobile Support returns data from the backend application to the device.

- Invoke any of the other methods in the Mobile Support Client library as needed for your business needs. For details on these methods, see *webMethods Mobile Support Client Java API Reference*.

## Writing the Logic to Initiate Synchronization Requests

If the mobile device is intended to send as well as receive data, your mobile application should include logic to do the following:

- Upload changed data from the device to the backend application.
- Download any data from the backend application that has changed since the mobile application last submitted a synchronization request.

Include the following:

- Ensure that the structure of the data to be passed to the sync method matches the format specified in `RowStructure`. Within the RowStructure class, invoke the setDataRow method to identify the business data, and invoke the setUniqueId method to set the unique ID for the business data.
- Invoke the sync method from the com.softwareag.mobile.data.sync.DataSynchronization class. As input to this method, pass the mobile sync component alias and the `RowStructure` parameters containing the business data to be synchronized. You can obtain the mobile sync component alias from the business integration developer who created the mobile sync component.

  **Note:** If you are using a filter in the download method to specify subsets of data to download from the backend application, the Mobile Support Client uses the same filter to process the sync method. If you change the filter key or value, you must invoke the download method again to refresh the device's local database before you invoke the sync method.

- The com.softwareag.mobile.data.sync.ResponseSet class encapsulates the response from the synchronization request. This class includes responses for all records sent from the mobile application, as well as updated records from the server (filtered, if a filter was supplied with the request). Use the getRowStructures and getStatus methods in this class to process the response further.

  Mobile Support returns the following status codes:

| Status Code | Meaning |
|---|---|
| `30` | An error occurred in the configuration or state of the mobile sync component associated with the mobile application.<br><br>To resolve this error, work with the server administrator to ensure that the association between the mobile application version and the mobile sync component is valid and that the mobile sync component is enabled. If appropriate for your business need, have the mobile application provide the user with feedback about the error and instructions for resubmitting the data. |
| `31` | The mobile application passed a filter with the request, and a filter validation error occurred.<br><br>To resolve this error, ensure that the mobile application passes the expected number of filter keys and that the filter key names match those specified in the mobile sync component's filter definition. |
| `40` | Mobile Support processed the synchronization request successfully. |

| Status Code | Meaning |
|---|---|
| | Overwrite the data in the mobile device's local database with the new data received from the server.<br><br>**Note:** Overwrite the mobile device's local database only when the server returns a status code of `40`. |
| `50` | Mobile Support encountered an internal error while processing the synchronization request.<br><br>Work with the server administrator to resolve this error. If appropriate for your business need, have the mobile application provide the user with feedback about the error and instructions for resubmitting the data. |

- Invoke any of the other methods in the Mobile Support Client library as needed for your business needs. For details on these methods, see *webMethods Mobile Support Client Java API Reference*.

## Writing the Logic for Logging Activities

Your mobile application should contain logic for logging activities. Invoke the getLogs method from the com.softwareag.mobile.data.sync.DataSynchronization class to retrieve entries from the log table for debugging purposes.

# Step 4: Associate the Mobile Application with a Mobile Sync Component

**Who performs:** Business integration developer

**Development tools needed:**webMethods Integration Server Administrator

After you create the mobile application, you associate the application with the mobile sync component you created in "Step 2: Create a Mobile Sync Component" on page 26.

## Associating a Mobile Application with a Mobile Sync Component

After you write the mobile application, you must associate the application with a mobile sync component and then enable the component so that Mobile Support can start processing data synchronization requests from the application.

**To associate a mobile application with a mobile sync component**

1. Open Integration Server Administrator if it is not already open.

2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Applications**.
4. Click **Add Mobile Application**.
5. On the Add Mobile Application screen, provide the following information:

| For this parameter... | Specify... |
|---|---|
| **Application Name** | The name of the mobile application to associate with the mobile sync component. |
| **Application Version** | The version of the mobile application. |
| **Mobile Sync Component Alias** | The alias of the mobile sync component.<br><br>**Note:** If the mobile application will synchronize with more than one type of dataset, you must create a mobile sync component for each dataset type. For details about associating the mobile application with multiple mobile sync components, see "Associating a Mobile Application with a Different Mobile Sync Component" on page 36. |

6. Click **Save Changes**.
7. If the mobile sync component is not enabled, enable it. For details, see "Enabling a Mobile Sync Component" on page 37.

## Associating a Mobile Application with a Different Mobile Sync Component

You can associate a mobile application with multiple mobile sync components. For example, in a retail scenario where order details and inventory details are maintained in two separate datasets, you might want to associate the mobile application with two mobile sync components: one that handles synchronization requests for orders, and one that handles inventory synchronization requests.

Use this procedure to specify which mobile sync components to associate with a mobile application.

**To associate a mobile application with a different mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Applications**.

4. In the list of mobile applications, click the name of the application with which you want to associate a different mobile sync component.
5. From the **Mobile Sync Component Alias** list, select one or more mobile sync components to associate with the application.
6. Click **Save Changes**.

# Managing Mobile Sync Components

After you create mobile sync components and associate mobile applications with those components, you may need to perform several administrative tasks for the components while developing and testing your mobile data synchronization solution. These tasks include enabling, disabling, editing, suspending, resuming, and deleting the mobile sync components.

## Enabling a Mobile Sync Component

Mobile sync components are disabled when they are first created. Mobile Support cannot start processing data synchronization requests between a mobile client and the backend application until the associated mobile sync component is enabled.

When you enable a mobile sync component, Mobile Support initializes the sync store associated with that component by calling the download flow service with the *fetchAll* parameter set to Yes. If the mobile sync component is configured to not store business data in the sync store, Mobile Support creates entries in the sync store that correspond to the data but does not copy the business data itself.

**To enable a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. On the Mobile Sync Components screen, click the **No** link in the **Enabled** column of the mobile sync component you want to enable.

## Disabling a Mobile Sync Component

You can disable a mobile sync component. When you do so, Mobile Support does the following:

- Stops processing data synchronization requests between a mobile application and a backend application
- Clears the contents of the associated sync store

**To disable a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. On the Mobile Sync Components screen, click the **Yes** link in the **Enabled** column of the mobile sync component you want to disable.

## Editing a Mobile Sync Component

You can edit a mobile sync component to specify a different download or upload service, specify a different business document type or different row identifiers, change filter keys, or change the conflict resolution rule that Mobile Support applies to any new synchronization requests the server receives. You can also change whether Mobile Support stores business objects in the sync store. For requests that were still in progress when the mobile sync component was changed, Mobile Support will apply the parameters that were in place when those requests were submitted.

You can edit a mobile sync component only when that component is disabled.

**To edit a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. If the **Enabled** column to the right of the mobile sync component contains **No (Suspended)** or **Yes**, do one of the following to disable the component:
   - If the mobile sync component is enabled, click the **Yes** link in the **Enabled** column.
   - If the mobile sync component is suspended, click the **Clear Store** link in the **Action** column to clear the sync store and disable the component.
5. In the **Alias** column, click the alias of the mobile sync component.
6. On the Edit Mobile Sync Component screen, change the following parameters as desired:

| For this parameter... | Specify... |
|---|---|
| **Download Service** | The fully qualified name of the flow service created to return data from the backend application to the mobile device. |
| **Upload Service** | The fully qualified name of the flow service created to send data from the mobile device to the backend application. |

| For this parameter... | Specify... |
| --- | --- |
| **Business Doc Type** | The fully qualified name of the IS document type to use to define the structure of the data to be synchronized. |
| **Row Identifiers** | One or more field names from the business document type that you want to use to uniquely identify the record to be synchronized. Separate field names with commas. |
| **Filter** | Optional. One or more keys that specify which records to return to the mobile application. Separate key names with commas.<br><br>The filter keys you specify here must be present in the business document type, and the mobile application should send these same keys in the synchronization request.<br><br>Mobile Support sends the requesting mobile application only the set of records that matches the filter, as follows:<br><br>■ If more than one filter key is passed, Mobile Support returns only those records that match all of the specified filter keys.<br><br>■ If a filter key contains more than one value, Mobile Support returns records that match any of the values. |
| **Conflict Resolution Rule** | The rule that Mobile Support should use when determining how to resolve conflicts that arise with new synchronization requests. Select either **Client wins** or **Server wins**.<br><br>For more information about conflict resolution rules, see "Change Detection and Conflict Resolution" on page 13. |
| **Store Business Data** | Whether to store, in the sync store, the business data to be synchronized. Recommended settings are as follows:<br><br>■ If the backend data is updated only by mobile applications that submit synchronization requests to Mobile Support, clear this check box.<br><br>■ If the backend data is updated by the backend application, outside of Mobile Support, as well as by mobile applications that submit synchronization requests to Mobile Support, select this check box. Doing so assists Mobile Support in detecting and resolving data conflicts. |

7. Click **Save Changes**.
8. Enable the mobile sync component. For instructions, see "Enabling a Mobile Sync Component" on page 37.

## Suspending a Mobile Sync Component

Certain situations can interfere with the processing of a synchronization request on the server, such as if the sync store becomes inaccessible, when the specified download or upload service is unavailable, or when a package that contains the download or upload service or the business document type is disabled or reloaded. In these situations, Mobile Support does one of the following to temporarily block synchronization requests and notifications associated with that mobile sync component:

- If the mobile sync component is configured to store business data in the sync store, Mobile Support suspends the mobile sync component.
- If the mobile sync component is configured to not store business data in the sync store, Mobile Support disables the mobile sync component.

You can also manually suspend a mobile sync component if that component is configured to store business data in the sync store. If the component is not configured to store business data, you must disable the component instead.

**Note:** If the WmMobileSupport package is disabled or reloaded, the mobile sync components remain in the same state they were in (enabled, disabled, or suspended) when the package was active.

When a mobile sync component is suspended, Mobile Support does the following:

- Completes synchronization requests received before the mobile sync component was suspended
- Rejects new requests
- Retains the contents of the associated sync store at the time the mobile sync component was suspended

**To suspend a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. On the Mobile Sync Components screen, click the **Suspend** link in the **Action** column next to the mobile sync component you want to suspend.

## Resuming a Mobile Sync Component

You can resume a suspended mobile sync component. When you do so, Mobile Support refreshes the associated sync store with changes made to the backend data while the mobile sync component was suspended. Mobile Support also sets the status of the mobile sync component to Enabled. Alternatively, you can disable a suspended mobile sync component, which clears the contents of the associated sync store so that you can start with a fresh set of backend data the next time the mobile sync component is enabled.

The mobile sync component specifies the download and upload service to execute and the business document type to use. If the package containing these elements is disabled or reloaded, Mobile Support suspends the mobile sync components that specify those elements. Follow the steps in this procedure to resume the suspended mobile sync components.

Resuming a mobile sync component is applicable only when the component is configured to store business data in the sync store.

**To resume a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. In the Navigation panel, click **Mobile Sync Components**.
4. On the Mobile Sync Components screen, do one of the following:
   - To resume a suspended mobile sync component, on the Mobile Sync Components screen, click the **No (Suspended)** link in the **Enabled** column next to the mobile sync component.
   - To disable a suspended mobile sync component and clear the contents of its associated sync store, click the **Clear Store** link in the **Action** column next to the mobile sync component.

## Deleting a Mobile Sync Component

You can delete a mobile sync component if that component is disabled or suspended and if the component is not associated with any mobile applications.

**To delete a mobile sync component**

1. Open Integration Server Administrator if it is not already open.
2. In the **Solutions** menu of the Navigation panel, click **Mobile Support**.
3. Remove any associations between the mobile sync component and mobile applications as follows:
   a. In the Navigation panel, click **Mobile Applications**.

b. On the Mobile Applications screen, check the Mobile Sync Component Aliases column. If the mobile sync component you want to delete is listed in that column, click the name of the application with which it is associated.

c. On the Edit Mobile Applications screen, clear the mobile sync component from the list of mobile sync component aliases, and then click **Save Changes**.

4. In the Navigation panel, click **Mobile Sync Components**.

5. If the mobile sync component you want to delete is enabled, do the one of the following:

   - Disable the mobile sync component by clicking the **Yes** link in the **Enabled** column for that component.
   - Suspend the mobile sync component by clicking the **Suspend** link in the **Action** column next to the mobile sync component. This option only applies when the mobile sync component is configured to store business data in the sync store.

6. Click the **Delete** button to the right of the mobile sync component you want to delete.

# 4 Frequently Asked Questions

# Questions about Mobile Support and Integration Server Quiesce Mode

This section answers questions about Mobile Support operations when the Integration Server that hosts Mobile Support enters quiesce mode. For more information about quiesce mode, see *webMethods Integration Server Administrator's Guide*.

- **What happens to the processing of synchronization requests when the host Integration Server enters quiesce mode?**

  If Mobile Support has already started processing a synchronization request, Mobile Support attempts to complete the processing of the request within the specified quiesce timeout period before Integration Server starts disabling packages.

  While Integration Server is in quiesce mode, new requests are ignored and must be resubmitted after the server exits quiesce mode.

- **What happens to mobile sync components when the host Integration Server enters and exits quiesce mode?**

  All mobile sync components retain their state when the server enters quiesce mode.

  While Integration Server is in quiesce mode, all notifications sent to Mobile Support for enabled mobile sync components are lost. The action that Mobile Support takes for these components when the server exits quiesce mode depends on how the components are configured. If a component is configured to store business data in the sync store, Mobile Support refreshes the sync store with the latest data from the backend application. If the component is configured to not store business data in the sync store, Mobile Support does not refresh the corresponding sync store entries; to refresh the sync store with changes made to the backend data while the server was in quiesce mode, you must disable and then enable the mobile sync component.

# Questions about Mobile Support Operations when Packages Are Disabled or Reloaded

This section answers questions about Mobile Support operations when the Integration Server packages that Mobile Support uses are disabled or reloaded.

- **What happens if the package containing the business document type or the download and upload services is disabled or reloaded?**

  Mobile Support handles the mobile sync components that refer to those elements according to how the components are configured. If a mobile sync component is configured to store business data, Mobile Support suspends the component. If the component is configured to not store business data, Mobile Support disables the component.

- **What happens if the WmMobileSupport package is disabled or reloaded?**

  The mobile sync components remain in the same state they were in (enabled, disabled, or suspended) when the package was active.

## Questions about Storing Business Data in the Sync Store

This section answers questions about storing business data in the Mobile Support sync store.

- **When is it necessary to store business data in the sync store?**

  Software AG recommends storing business data in the sync store when changes are made to the backend data not only by mobile applications that submit synchronization requests to Mobile Support but also by the backend application itself. Doing so assists Mobile Support in detecting and resolving data conflicts.

- **When is it not necessary to store business data in the sync store?**

  It is not necessary to store business data in the sync store when changes are made to the backend data only by mobile applications that submit synchronization requests to Mobile Support.

  In certain situations, you might opt to not store business data in the sync store even if the backend application updates the backend data outside of Mobile Support. These situations depend on your business need. Examples of these situations include the following:

  - The client subscribes to data from the backend application but does not publish data back to the server.
  - The backend application updates the backend data during a time when no synchronization activity is expected from mobile devices, such as during off-peak hours.
  - A task is assigned to a specific user who submits synchronization requests from a particular device. No other users will be updating the same set of backend data, and the assigned user will submit requests from a single device (for example, only from his or her tablet, or only from his or her mobile phone).

# A Built-In Services

# Overview

webMethods Mobile Support includes a built-in service and two specifications that facilitate data synchronization in a mobile application environment.

# wm.mobile.datasync:synchronize

Synchronizes data between a mobile device and a backend application.

This service is used for the following purposes:

- On the client side, a mobile application invokes this service by way of the Mobile Support Client library to send data to, and receive data from, the backend application through Mobile Support.
- On the server side, the business integration developer invokes this service within a notification service. The notification service alerts Mobile Support that the backend application has updated the backend data directly, outside of Mobile Support, and the synchronize service synchronizes the sync store with the changes the backend application made.

Mobile Support determines whether a synchronization request comes from a mobile device or from a backend notification service based on the set of parameters provided in the synchronization request message. When the request header contains a mobile application name and a mobile application version, this indicates that the request is from a mobile device. If this information is not present, Mobile Support identifies the request as coming from a notification service.

### Input Parameters

*inputParams* **Document** Parameters that specify the metadata that Mobile Support uses to track change requests.

| Key | Description |
|---|---|
| *mscAlias* | **String** Alias used to identify the mobile sync component associated with this request. The alias name is specified during creation of the mobile sync component.<br><br>This parameter is required when the synchronize service is invoked either from the Mobile Support Client library or from a backend notification service. |

*lastVersion* **String** Optional. Last version of the data processed in a synchronization request associated with the specified mobile sync component.

**Note:** If a mobile application is submitting a synchronization request on behalf of a mobile device for the first time, *lastVersion* will be blank. This indicates to Mobile Support to treat the request the same as a download request.

This parameter is not required when the synchronize service is invoked from a backend notification service.

*filter* **Document** Optional. List of key/value parameters, if passed by the mobile application, that specify the subset of records to download to the mobile device.

- If more than one filter key is passed, Mobile Support returns only those records that match all of the specified filter keys.
- If a filter key contains more than one value, Mobile Support returns records that match any of the values.

Mobile Support uses this parameter internally to pass the filter from the Mobile Support Client to the server.

This parameter is not used when the synchronize service is invoked from a backend notification service.

*inputPayload* **Document List** Data to be synchronized.

| Key | Description |
|---|---|
| *mDataRow* | **Object** The record to be synchronized, in IData format.<br><br>If the mobile sync component associated with this request is configured to not store business data in the sync store, the data in this variable will be passed to Mobile Support but not stored in the sync store. |

If you are calling the synchronize service from a notification service that processes delete requests, include, at a minimum, the fields defined as row identifiers. For insert and update requests, include all fields specified in the business document type.

*uid* **String** Optional. A unique identifier for the record. Mobile Support constructs the unique ID using the row identifiers supplied when the corresponding mobile sync component was created.

Mobile Support returns the unique ID as part of the synchronize service's output signature for each record processed. For subsequent update and delete requests for a particular record, be sure to use the most recent unique ID that the server returned for that record.

This parameter is not required when the synchronize service is invoked from a backend notification service.

*actionId* **String** Action to be taken on a particular record. A value of:

- `0` indicates to insert the record.
- `1` indicates to delete the record.
- `2` indicates to update the record.

### Output Parameters

*outputParams* **Document** Status of the request.

| Key | Description |
|---|---|
| *currentVersion* | **String** Conditional. Current version of the data associated with the specified mobile sync component.<br><br>Mobile Support does not return a value for this parameter when the synchronize service is called from a backend notification service. |

*statusCode* **String** Indicator that depicts the status of the request. A value of:

- `30` indicates that an error occurred because the association between the mobile application version and the mobile sync component is not valid, or the associated mobile sync component is suspended or disabled, or Mobile Support was unable to derive the unique ID from the record.
- `31` indicates that the mobile application either passed filter keys that did not match those defined in the mobile sync component or did not pass the expected number of filter keys.
- `40` indicates that Mobile Support processed the synchronization request successfully.
- `50` indicates that Mobile Support encountered an internal error while processing the synchronization request.

Mobile Support returns a status code for each request that reaches Mobile Support. Mobile Support may not be able to return a status code if the synchronize service cannot be invoked, for example if Integration Server is down or is being restarted.

*outputPayload* **Document List** Data to be sent back to the application.

| Key | Description |
|---|---|
| *mDataRow* | **Object** A single record returned by Mobile Support, in IData format.<br>For delete requests, this object will be empty. |
| *uid* | **String** A unique identifier for the record. Mobile Support constructs the unique ID using the row identifiers supplied when the corresponding mobile sync component was created. |
| *actionId* | **String** Action taken on a particular record. A value of:<br>- `0` indicates the record was inserted. |

- `1` indicates the record was deleted. For delete requests, the *mDataRow* object will be empty.
- `2` indicates the record was updated.

*rowStatus* **String** Status of the record transaction. A value of:

- `0` indicates that Mobile Support was successful in processing the requested action.
- `1` indicates that Mobile Support failed to process the request.
- `2` indicates that Mobile Support detected a conflict and resolved the conflict according to the conflict resolution rule in effect. Mobile Support sends the resolved data back to the application.
- `3` indicates that no operation was performed because Mobile Support detected a conflict between the sync store and the backend data and did not have enough information to process the request further. The synchronization request must be resubmitted.

### Usage Notes

- Mobile Support does not consider the execution of the synchronize service to be unsuccessful if the corresponding adapter service fails. Therefore, handle any adapter service exception situations within the calling flow service.
- Mobile Support uses the IS document type you created in "Step 1: Write the Server-Side Business Logic" on page 20 to define the structure and contents of the *mDataRow* variable.

## wm.mobile.datasync.specs:downloadSpec

Specification for the flow service used to return data to a mobile device.

### Input Parameters

*inputArgs* **Document** Optional. Contains the fields specified as row identifiers in the business document type.

*fetchAll* **String** Specifies how much data to return to the mobile device. A value of:

- `Yes` indicates to return all data. This is the default when Mobile Support detects that this is the first time that a synchronization request was submitted from this mobile device.
- `No` indicates to return only the record specified in *inputArgs* .

In either case, if the mobile application specified a filter with the request, Mobile Support returns only the data that matches the filter criteria.

### Output Parameters

*outputSyncData* **Document List** List of records received from Mobile Support.

| Key | Description |
|---|---|
| *mDataRow* | **Object** A single record received from Mobile Support, in IData format.<br><br>Mobile Support uses the IS document type created in "Step 1: Write the Server-Side Business Logic" on page 20 to define the structure and contents of the *mDataRow* variable. |

## wm.mobile.datasync.specs:uploadSpec

Specification for the flow service used to send data to Mobile Support.

### Input Parameters

*inputSyncData* **Document List** List of records to send to Mobile Support.

| Key | Description |
|---|---|
| *mDataRow* | **Object** A single record to send to Mobile Support, in IData format. |
| *actionId* | **String** Action to be taken on a particular record. A value of:<br><br>- `0` indicates to insert the record.<br>- `1` indicates to delete the record. |

- `2` indicates to update the record.

### Output Parameters

*outputSyncData* **Document List** List of records sent to Mobile Support.

| Key | Description |
|---|---|
| *mDataRow* | **Object** A single record sent to Mobile Support, in IData format.<br>Mobile Support uses the IS document type created in "Step 1: Write the Server-Side Business Logic" on page 20 to define the structure and contents of this variable. |
| *rowStatus* | **String** Status of the record transaction. A value of:<br>- `0` indicates that Mobile Support was successful in processing the requested action.<br>- `1` indicates that Mobile Support failed to process the request. |
| *actionId* | **String** Action taken on a particular record. A value of:<br>- `0` indicates the record was inserted.<br>- `1` indicates the record was deleted.<br>- `2` indicates the record was updated. |