TIBCO ActiveMatrix BusinessWorks™ Plug-in for Mobile Integration User's Guide

Software Release 6.2.0
December 2016

Two-Second Advantage®

TIBCO®

Important Information

Contents

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

https://docs.tibco.com

Product-Specific Documentation

The following documents for this product can be found in the TIBCO Documentation Library:

- TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration Installation
- TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration User's Guide
- TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration Developer's Guide
- TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration Expresso Server Guide
- TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration Release Notes

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

https://www.tibcommunity.com

About TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration

TIBCO ActiveMatrix BusinessWorks™ Plug-in for Mobile Integration (BWMI) enables sending push notifications to Apple iOS and Google Android applications from TIBCO ActiveMatrix BusinessWorks™ Enterprise processes.

With tremendous growth of smart phones and tablets, the mobility sector is getting at the core of IT strategy. It has become increasingly important to integrate mobile applications into business processes. TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration provides a solution to this market demand. It integrates TIBCO ActiveMatrix BusinessWorks Enterprise with mobile platforms by:

- Allowing the TIBCO ActiveMatrix BusinessWorks Enterprise processes to send push notifications to a mobile device in a reliable way.
- Allowing mobile applications to send device token and user information back to the server. This helps map the mobile application on the mobile device with business user information.
- Allowing mobile applications to acknowledge notification receipt and also pull pending notifications by connecting back to the Mi Service running in the TIBCO ActiveMatrix BusinessWorks Enterprise server.

To receive push notifications on your mobile device from an application provider:

1. Install a mobile application from that particular application provider.
2. The application must register itself with the notification service and receive a device token. The device token denotes the address of the application running on that mobile device. It identifies both the application and the device it is running on.

   

   Before you start your mobile notifications development, do consider that the device tokens may change as per the platform vendor's guidelines.
3. The application sends this device token and any other user-specific information to the application provider's server.

The application provider can now send notifications to the mobile device on which the application was installed. This is done through the platform providers' push notification service using the device token.

TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration Components

TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration consists of 2 main components. They are MI Service and Plug-in for Mobile Integration.

MI Service

The MI service component running inside the TIBCO ActiveMatrix BusinessWorks Enterprise engine, offers the following key functionalities:

- Communicates with the push notification service (APNS and GCM) to send out push notifications in a scalable manner.
- Has an in-memory notification store to manage notification retries, timeout, and badge updates for the reliable notifications.
- Provides REST services for the mobile applications to register device token, and acknowledge and pull notifications from the mobile applications.
- Generates appropriate events to trigger different BWMI event sources based on the state of notifications.

TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration

TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration comprises of the following activities, event sources, and signal-ins:

- Acknowledgement Receiver
- Feedback Receiver
- Send Push Notification
- Subscription Receiver
- Timeout Receiver
- Wait for Acknowledgement
- ExpressoNotify

This plug-in also comprises of the following shared resources:

- Mi Service
- PNS Communicator
- HTTP Connector
- Expresso Provider

MI Palette

You can use the MI Palette to create the BusinessWorks processes to send push notifications to the mobile devices and integrate TIBCO ActiveMatrix BusinessWorks Enterprise with mobile platforms.

Acknowledgement Receiver

The Acknowledgement Receiver event source gets an acknowledgment received at Mi Service for the push notification received by the mobile application. It initiates any existing TIBCO ActiveMatrix BusinessWorks Enterprise process for the user to handle the processing of the received acknowledgments.

General

The **General** tab has the following fields.

Field	Description
Name	The name to be displayed as the label for the process.

Description

Provide a short description of the process.

Output

The following is the output for the activity.

Output Item	Description
notificationID	The ID of the acknowledged notification.
deviceToken	The device token of the mobile application acknowledging the notification.
businessId	Optional. Use this field to enable group notifications for processing the messages.  TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration does not define the business ID.

Feedback Receiver

The Plug-in for Mobile Integration receives a feedback from the APNS or the GCM server where the device declined to accept the notification because the target application on the device ceased to exist. Plug-in for Mobile Integration gathers feedback from the APNS or GCM server and provides feedback through this event source.

This activity triggers when the:

- Deregistration service is called,
- APNS feedback poller returns invalid IDs, and
- GCM server returns invalid IDs

General

The **General** tab has the following fields:

Field	Description
Name	The displayed name of the process.

Description

Provide a short description of the process.

Advanced

The **Advanced** tab has the following fields.

Field	Description
Sequence Key	This field can contain an XPath expression that specifies which processes should run in sequence. Process instances with sequencing keys that evaluate to the same value are executed sequentially in the order the process instance was created.
Custom Job Id	This field can contain an XPath expression that specifies a custom ID for the process instance.

Output

The following is the output for the activity.

Output Item	Datatype	Description
feedbackEntry	text	The feedback of the application status when unregistered.
failedDeliveryTime	number	The exact time when the delivery failed, as reported by the APNS or GCM server.
deviceToken	complex	The device token for which the delivery failed.

Send Push Notification

You can use this activity to create notification and push this notification to the MI Service.

General

The **General** tab has the following fields.

Field	Description
Name	The name to be displayed as the label for the activity in the process.

Description

Provide a short description for the activity.

Input

This activity has the following input.

Input Item	Description
alertText	The text used as an alert message.
sound	Name of the sound file to be played, when the notification is delivered on the device.
badge	Signifies number of waiting notifications on the server. This value is used only for **fireAndForget** type of notifications. For managed notifications the badge number is: • dynamically stamped by TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration, and • the value is equal to the number of unacknowledged notifications for the specified device token at the time of dispatch.
businessId	An optional custom id to be associated with a set of notifications to be used by the business users.
collapseKey	Collapse key is added to the outgoing message. When multiple messages are queued up in the Mi Service for the same user and collapseKey is set, only the last one with any specified collapse key is delivered. For details, refer to Collapse Key.
devices	High level group to hold platform specific device identifiers. Currently, iOS and Android devices are supported.
fireAndForget	Optional. If set to `true`, the TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration neither tracks the notification delivery, nor awaits the acknowledgement from the device. Fire and Forget notifications are not managed by this plug-in. The delivery of such notifications is governed by the Quality of Service (QoS) of the respective platform notification service. The default value is set to `false`.
retryCount	Specify for the number of times the server can retry to send the notification, if the device does not respond by sending an acknowledgement within a stipulated time. Actual retry is limited by the **timeToLive** value.  Retry works only for live notifications.  Default value is set to 0. The negative values if specified, gets converted to 0. Specify only a +ve value, that is, 1 or greater.

Input Item	Description
retryInterval	Specify to set the time interval (in milliseconds) between the retry process of the notifications.  Default value is set to 0. The negative values if specified, get converted to 0. Specify only a +ve value, that is, 1 or greater.
timeToLive	Specify to set the time limit (in minutes) for the notification to be retained by the Mi Service.  The default time is set to "5" minutes.

Output

The following is the output for the activity

Output Item	Description
notificationId	The unique ID of the notification generated by the Mi Service.

Subscription Receiver

Subscription Receiver activity receives activation and deactivation requests for receiving the notification from the mobile device. This activity triggers, when the application subscribes or unsubscribes the device token. The mobile application is responsible to notify Mi Service of its subscription status by calling its subscription service when appropriate.

General

The **General** tab has the following fields.

Field	Description
Name	The name to be displayed as the label for the activity in the process.

Description

Provide a short description for the activity.

Advanced

The **Advanced** tab has the following fields.

Field	Description
Sequence Key	This field can contain an XPath expression that specifies which processes should run in sequence. Process instances with sequencing keys that evaluate to the same value are executed sequentially in the order the process instance was created .
Custom Job Id	This field can contain an XPath expression that specifies a custom ID for the process instance.

Output

The following is the output for the activity.

Output Item	Datatype	Description
Subscription	text	This element consists of the following: • status: of the mobile application whether it should receive push notifications or not. • deviceToken: Unique ID of the application running on the device. • platformType: type of mobile platform the device is operating on.
user	text	Contains the following user-specific information: • phone: contact number of the user • email: email ID of the user

Timeout Receiver

Timeout Receiver activity receives the timeout event when the managed notifications expire. Expiration means that at least one or more devices that were supposed to receive the notification, did not acknowledge the receipt of the notification. This activity triggers, when the notifications expire on the server.

General

The **General** tab has the following fields.

Field	Description
Name	The name to be displayed as the label for the activity in the process.

Description

Provide a short description for the activity.

Advanced

The **Advanced** tab has the following fields.

Field	Description
Sequence Key	This field can contain an XPath expression that specifies which processes should run in sequence. Process instances with sequencing keys that evaluate to the same value are executed sequentially in the order the process instance was created.
Custom Job Id	This field can contain an XPath expression that specifies a custom ID for the process instance.

Output

The following is the output for the activity.

Output Item	Datatype	Description
TimeOutNotification	text	This element consists of the following: • **notificationId**: The unique ID of the notification generated by the Mi Service. • **alertText**: The text used as an alert message. • **badge**: The number of waiting notifications on the server. For managed notifications the badge number is: – dynamically stamped by TIBCOActiveMatrix BusinessWorks Plug-in for Mobile Integration, and – the value is equal to the number of unacknowledged notifications for the specified device token at the time of dispatch. • **sound**: Name of the sound file to be played, when the notification is delivered on the device. • **businessId**: An optional custom id to be associated with a set of notifications to be used by the business users. • **collapseKey**: Collapse key added to the outgoing message. For details, refer to Collapse Key. • **creationTime**: The creation time of the notification. • **expirationTime**: The time when the notification expires on the server. • **timeToLive**: The time limit (in minutes) for the notification to be retained by the Mobile Integration Service.  The default time is set to "5" minutes. • **retryCount**: The number of times the server can retry to send the notification, if the device does not respond by sending an acknowledgement notification within a stipulated time. Actual retry is limited by the **timeToLive** value.  Retry works only for live notifications.  Default value is set to 0. The negative values if specified, gets converted to 0. Specify only a +ve value, that is, 1 or greater. • **retryInterval**: Specified time interval (in milliseconds) between the retry process of the notifications.  Default value is set to 0. The negative values if specified, get converted to 0. Specify only a +ve value, that is, 1 or greater.

Output Item	Datatype	Description
acknowledgedDevices	text	Contains the list of device tokens for the iOS and Android devices and the acknowledged receipt of the notification by the user.
unacknowkledgedDevices	text	Contains the list of device tokens for the iOS and Android devices and the unacknowledged receipt of the notification by the user.

Wait for Acknowledgement

This activity is for the inline use within the business processes. The activity suspends the business process until all the devices, to whom the notification was sent, acknowledge the receipt. If the notification is not acknowledged within a specified timeout value, it raises a time-out exception.

General

The **General** tab has the following fields.

Field	Description
Name	The name to be displayed as the label for the activity in the process.

Description

Provide a short description for the activity.

Input

The activity has the following input.

Activity Input	Description
notificationId	The notification ID of the push notification.
timeout	The timeout of this activity.  It is recommended not to set the value as negative. Setting this value as negative, for example, `-50000ms`, throws an error.

ExpressoNotify

You can use the ExpressoNotify activity to generate events and send notifications to the Expresso server.

The activity includes an input editor where you can define the input data.

General

The **General** tab has the following fields

Field	Description
Name	The name to be displayed as the label in the process.

Description

Provide a short description for the activity.

Input Editor

The input editor tab defines the structure of the process variables to be added to the process. The input schema uses only primitive elements. Define a few attributes of the event data that are relevant to the mobile application UI.

You can also reference XML schemas (with primitive elements) that are stored in the project. The data specified in the input editor is the input for the ExpressoNotify activity.

Field	Description
Name	The name to be displayed as the label in the process.

Input

The input for the activity is defined by the data elements specified in the input editor tab.

Shared Resource

Shared configuration resources are specifications that are shared among activities. This plug-in comprises and uses Mi Service, PNS Communicator, and HTTP Connector shared resources.

MI Service

The MI Service shared resource collects notifications from the TIBCO ActiveMatrix BusinessWorks Plug-in for Mobile Integration and sends to the push notification service. This shared resource holds the MI service configuration.

General

The **General** tab has the following fields.

Field	Description
Name	Name of the MI service shared resource.
Description	Short description of the MI service shared resource.

MI Service

Specify appropriate values in the following fields.

Field	Description
Badge Update Interval (Mins)	This is the interval at which BWMI notifies all the mobile applications for whom there are pending notifications. BWMI sends a special notification called the badge update notification which contains the count of pending and unacknowledged notifications for a particular application. Default is 15 minutes.
Acknowledgement Processor Thread Pool Size	This size specifies the number of threads in the ACK processor thread pool and not the size of the ACK processor queue. The threads in the pool actually process the acknowledgements received from the mobile applications. The user can adjust the number of threads to be employed to process the incoming ACKs load.
Context Path	Specify the prefix of a URL path that is used to select the context(s) to which an incoming request is passed.

HTTP Connector

The HTTP Connector shared resource holds HTTP service configuration which the MI Service uses to deploy its rest services on. For details about HTTP Connector, refer to *TIBCO ActiveMatrix BusinessWorks Enterprise Palette Reference*.

Click the ... icon, and create a new **HTTPConnectorResource** from the Select HttpConnResource Resource Template window.

PNS Communicator

PNS communicator shared resource holds configuration related to the APNS and GCM services. For details refer to PNS Communicator.

Click the ... icon, and create a new **PNSCommunicatorResource** from the Select PNSCommunicatorResource Resource Template window.

PNS Communicator

PNS Communicator shared resource holds configurations related to the APNS and GCM services.

General

The **General** tab has the following fields.

Field	Description
Name	Name of the PNS communicator shared resource.
Description	Short description of the PNS communicator resource.

APNS Configuration

Specify appropriate values in the following fields.

Field	Literal Value/ Module Property	Description
Host Name	Yes	Host name or IP address of the APNS.
PNS Port	Yes	Apple Push Notification service port number. Default is 2195.
Feedback Port	Yes	Apple Push Notification Feedback service port number. Default is 2196.
Feedback Poll Interval (Mins)	Yes	Interval in minutes for polling the feedback service. Default is 15 minutes.
Dispatcher Thread Pool Size	Yes	Specifies the number of threads to be employed to push the notifications to APNS. Every thread maintains its own separate persistent connection to APNS.
Minimum Notification Expiry (Secs)	Yes	Minimum time to live threshold for the iOS notifications. Any value below this threshold will be automatically promoted to this threshold value. Default is 300 seconds.

APNS Certificates

Specify appropriate values in the following fields.

Field	Literal Value/ Module Property	Description
Trust Store Location	Yes	The Trust store contains trusted certificates. Provide the location of the Trust Store file.

Field	Literal Value/ Module Property	Description
Trust Store Password	Yes	The password to access the trust store file.
Key Store Location	Yes	The Key store contains private keys certificates. Provide the location of the Key Store file.
Key Store Password	Yes	The password to access the key store file.

GCM Configuration

Specify appropriate values in the following fields.

Field	Literal Value/ Module Property	Description
API Key	Yes	Specify the API key which the device will use to send requests to the GCM server.
Dispatcher Thread Pool Size	Yes	Specifies the number of threads to be employed to push the notifications to GCM.
Minimum Notification Expiry (Secs)	Yes	Minimum time to live threshold for the Android notifications. Any value below this threshold will be automatically promoted to this threshold value. Default is 300 seconds.

Expresso Provider

The Expresso Provider shared resource collects events from the ExpressoNotify activity and sends them to the Expresso server.

General

The **General** tab has the following fields.

Field	Description
Package	Displays the name of the process package that contains the package. This field is not editable.
Name	The name of the Expresso Provider shared resource. The name is dispalyed as **Domain Name** on the mobile application UI.
Description	A short description of the Expresso Provider shared resource.

Provider Configuration

Specify appropriate values in the following fields.

Field	Literal Value/ Module Property	Description
Provider Description	Yes	A short description of the provider. The description is displayed as **Domain Description** on the mobile application UI.
Provider Color		Select the pod color for the providers on the mobile application UI.

Expresso Server Details

Specify appropriate values in the following fields.

Field	Literal Value/ Module Property	Description
Expresso Host	Yes	The host name or IP address of the Expresso Server. Default host is localhost.
Expresso Port	Yes	The port of the Expresso server. The default value is 36136.

Test Connection

The Test Connection button tests the connection to the specified Expresso server. You can use this button to ensure that the server client is configured correctly.

Provider Server Configuration

HTTP Connector

The HTTP Connector shared resource holds the HTTP service configuration. The Expresso Provider uses the shared resource to deploy its REST services on (listening to subscribe or unsubscribe requests). For details about the HTTP Connector, refer to *TIBCO ActiveMatrix BusinessWorks Bindings and Palettes Reference*.

Click the  icon, and select the existing **HTTPConnectorResource** or create a new **HTTPConnectorResource** from the Select HttpConnResource Resource Template window.

To reach the internal providers on servers deployed inside the firewall, specify the public host and public port properties. Provider end point URL's are generated using the public host and port properties.

To override the host and port values use the following system properties:

- `bw.plugin.mi.expresso.public.host`
- `bw.plugin.mi.expresso.public.port`

Event Configuration

One Expresso event is equivalent to one ExpressoNotify activity.

Specify appropriate values in the following fields.

Column Name	Description
Event Name	The name of the Expresso Event. This name is dispalyed as the name of the pod on the mobile application UI.  The **Event Name** must be unique in the provider and must not contain spaces.
Implementing Process	The qualified name of the event.
	Opens the Select an Event from Process dialog box. All the available events are displayed. Select the required event. To create a new event, click the **Create New Event Process** button. For more details, refer to the section on**Create New Expresso Event Wizard**.
	Deletes the selected event from the event table.
	Opens the process implementation for the selected event.
	Validates events and ensures that all the events are available in the module.
	Reselectes event implementation. If you rename the ExpressoNotify activity in an event or in a process containing the event, use this button to reselect the event implementation. After the reselection, validate the event using by clicking the validate button.
Description	The description of the selected event. The description is displayed as the decription of the pod on the mobile application UI.
Usage	Event Usage - sample usage of the selected event. The Usage is displayed as the pod sample usage on the mobile application UI. Event Data- displays the tree structure of the event data for the selected event. You can drag and drop data from the Event Data tree to Event Usage.
Schema	JSON Draft 4 schema for the selected event. This field is not editable.

Create New Expresso Event Wizard

Selecting the **Create New Event Process** button opens the Create New Expresso Event wizard. Create a new process with the Timer and ExpressoNotify Activity, using this wizard.

Field	Description
Process Folder	Specify a folder for the process.
Package	Specify a process package.

Field	Description
Process Name	Specify the process name.
Modifiers	Specify the modifiers.

Configure Event Wizard

Configure granular event details using the Configure Event wizard. These configurations can be modified later using the **Description**, **Usage**, and **Schema** tabs.

Field Name	Description
Event Name	The name of the Expresso Event. The name of the event is displayed as the name of the pod on the mobile application UI.  The **Event Name** must be unique in the provider.
Process	The qualified name of the event. This field is not editable.
Event Subscription URL	The subscription url to be used by the Expresso server, to subscribe and unsubscribe to the events. This field is not editable.
Description	The description of the the selected event. The description is displayed as the decription of the pod on the mobile application UI.
Event Usage	The sample of the selected event on the mobile application UI.

Registering the Providers

After configuring the ExpressoNotify activity and the Expresso Provider shared resource, run the ActiveMatrix BusinessWorks application. The providers are now registered with the configured Expresso sever.

 Currently, persistent storage of subscribers is not supported.

The OSGI command **rproviders** lists all the registered providers.

For example:

```
<>@BWEclipseAppNode rproviders
Provider 1
Provider Name Workday
Expresso Server IP http://localhost:36136
Registered At 2014/11/07 12:16:06 IST
No of Events 2
```

Collapse Key

Mobile Integration Plug-in provides a unified support for collapse keys for both APNS and GCM. This feature allows a key known as **Collapse Key** to be attached to the notifications. If multiple notifications arrive in a sequence with same collapse key, over a period of time, the Mobile Integration plug-in collapses these notifications at its end. This means that all older notifications are discarded and only the latest notification are stored on the MI Service. When a device comes online and pulls the notifications from the server, it receives only the latest notification among the notifications bearing the same collapse key. However, if the collapse key is not set, both the new and old messages are stored for the future delivery.

The best scenario suited is, when the device is offline and does not receive older notifications already dispatched to its platform notification service. In case the device is online, then it may still receive notifications that were already dispatched from its platform notification service. Collapse key facilitates collapsing of the waiting and undelivered notifications.

A perfect example for this would be about sending live cricket score updates to the users. If the user didn't receive the previous score updates then it is not necessary to store all them. The user would be interested only in the latest updates.

When a user sends a notification to the mobile application, the MI Service stores the notification and pushes it to APNS and/or GCM for delivery. MI Service maintains a notifications queue for each device from which the device can pull the messages addressed to it. Hence, if there are multiple notifications addressed to a particular device, then these notifications get queued up in its notifications queue in the sequence they are received.

The collapse key collapses only the notifications bearing the same collapse key. If a user sends 10 notifications with the same collapse key (for example, "SCORE") and 2 notifications without any collapse key and they are not received by the device as the device was offline. Then when the device is back online, it will receive total 3 (1+2) notifications, provided none of the 3 notifications expire in the mean time.

GCM supports collapse key out of the box while APNS has no concept of collapse keys. APNS does not require collapse notifications because it stores only a single notification always (which is always the latest) per device. APNS discards the older notifications automatically. However, GCM can store up to 100 messages per device and attempts to deliver them to the device when it is online or connected. Hence, GCM needs collapse keys to filter irrelevent notifications. Now, GCM allows a maximum of 4 different collapse keys per device at a specified time. You can have more than 4 collapse keys but not at the same time. For details refer to Android documentation for GCM.