

# Business Scoped User Ids

- [Business-scoped user IDs | Developer Documentation](#)

# Business-scoped user IDs | Developer Documentation

## Business-scoped user IDs

Updated: Mar 31, 2026

This document received a major update on March 18, 2026. See the [changelog entry](#) for details. WhatsApp is launching usernames later in 2026.

Usernames are an optional feature for users and businesses. If a username is adopted by a WhatsApp user, their username will be displayed instead of their phone number in the app. Business usernames are not intended for privacy, however. If you adopt a business username, it will not cause your business phone number to be hidden in the app.

To support usernames, Meta will share a new backend user identifier called business-scoped user ID, or BSUID. BSUID uniquely identifies a WhatsApp user and is tied to a specific business.

This document describes how the addition of usernames will impact API requests, API responses, and webhook payloads. Additional changes to support usernames before the feature is made available will be recorded here.

**Any changes described in this document are subject to change.**

## User usernames

A user username is a unique, optional name that WhatsApp users can set in order to display their username instead of their phone number in the app. Usernames can be used in lieu of profile names when personalizing message content for individual users.

WhatsApp users are limited to 1 username, but are able to change them periodically. Changing a username does not affect the user's phone number or business-scoped user ID, and does not affect the user's ability to communicate with other WhatsApp users or businesses on the WhatsApp Business Platform.

User usernames have the same format restrictions as [business usernames](#).

Usernames are assigned to the `username` property in API responses and webhooks payloads. Once enabled, a WhatsApp user's username will appear in all incoming [messages](#) webhooks, and all

**delivered** and **read** [status messages](#) webhooks.

## Business-scoped user ID

BSUIDs will begin appearing in webhooks in early April 2026.

A BSUID is a unique user identifier that can be used to message a WhatsApp user when you don't know their phone number. BSUID will be assigned to the `user_id` parameter and appear in all

[messages webhooks](#), regardless of whether or not the user has enabled the username feature. BSUIDs are scoped to individual business portfolios. This means that any business phone number owned by a given portfolio can be used to message a BSUID scoped to the same portfolio, and attempts to message the BSUID using a phone number owned by a different portfolio will fail. BSUIDs will be:

generated automatically prefixed with the user's [ISO 3166 alpha-2](#) two-letter country code and a period, followed by up to 128 alphanumeric characters (for example, `US.13491208655302741918`) unique to each business portfolio-user pair ([business portfolios](#) were formerly known as Business Managers) regenerated if a user changes their phone number (which trigger a [system status messages webhook](#))

BSUIDs can be used to send any type of message except for one-tap, zero-tap, and copy code [authentication templates](#), which require user phone numbers.

When making API requests with BSUIDs, use the entire BSUID value: country code, period, and all alpha numeric characters. Omitting or changing the country code, period, or alpha numeric characters will cause your request to fail.

If you are a managed business with multiple business portfolios, and want to use BSUIDs that will work across all of them, see [Parent business-scoped user IDs](#).

## Parent business-scoped user IDs

If you are a managed business and want to link business portfolios, you can ask your Meta point-of-contact to check if you are eligible. If you are eligible, and your business portfolios become linked, parent BSUIDs will be included in all messages webhooks, assigned to a new `parent_user_id` property.

Parent BSUIDs can be used in place of regular BSUIDS to message users. Functionally, parent BSUIDs have the same properties as regular BSUIDs, but can be used by any business phone number within the set of linked portfolios. Parent BSUIDs follow the same format as regular BSUIDs, but include `ENT` between the country code and the alphanumeric identifier (for example, `US.ENT.11815799212886844830`).

Note that you can still message users using their regular BSUID scoped to your business portfolio.

## Phone numbers

If a WhatsApp user enables the username feature, their phone number will not be included in webhooks, unless you have interacted with the user before, as explained below. Therefore, regardless of whether or not the user has enabled the feature, the user's BSUID will be included in any webhooks that would normally include their phone number, assigned to a new `user_id` property.

To reduce the chance of losing conversation context with existing users who enable the usernames feature, user phone numbers will be included in webhooks if any of the following conditions are met:

You have messaged or called the user's phone number within the last 30 days of the webhook being triggered  
You have received a message or call from the user's phone number within the last 30 days of the webhook being triggered  
The user is in your [contact book](#)

Note that the 30-day lookback conditions above are evaluated per business phone number. If you message a user from one of your business phone numbers, webhooks associated with a different business phone number in your portfolio will not include the user's phone number unless that specific number has also sent or received a message or call to or from the user's phone number within the last 30 days.

BSUIDs will begin appearing in webhooks in early April 2026. However, our APIs will not support sending messages targeted to the BSUIDs until May 2026 (exact date pending). Once our APIs support BSUIDs in May, you will be able to message users using either their BSUID, phone number, or both.

If you are a solution provider and provide WhatsApp messaging services to your business customers, your customers will be able to use your app to message users, using their portfolio's business phone numbers and any BSUIDs scoped to their portfolio. If you attempt to use one of your business customer's BSUIDs with your own business phone number, however, it will fail, since BSUIDs are scoped to portfolios (and essentially, the assets the portfolio owns).

If you are unsure of asset ownership:

Send a GET request to the [Client WhatsApp Business Accounts API](#) to get a list of WABAs that you do not own, but that are shared with you. Send a GET request to the [Owned WhatsApp Business Accounts API](#) to get a list of WABAs that you own. Send a GET request to the [Phone Numbers API](#) to get a list of phone numbers owned by a given WABA.

## Contact book

In early April 2026, to support messaging thread continuity, a contact book feature that stores WhatsApp user contact information is being released. The contact book is provided and hosted by Meta; no integration work is required.

Once the feature is available, if you send a message/call to a user's phone number, or receive a message/call from a user's phone number, the user's phone number and BSUID will be added to your contact book. Once this data has been recorded, it will be used to populate any webhook payloads and API responses that include the user's phone number or BSUID, regardless of whether or not the user has enabled the usernames feature.

The contact book is scoped to the business portfolio level, so any interaction between any business phone number within the business portfolio and a user will trigger the user's phone number and BSUID to be stored in the contact book. Only interactions that occur after the contact book launches will trigger storage; prior interactions will not be retroactively captured, and contact information from those users will not be included in API responses or webhooks.

Contact book data will be retained until you disable the feature, or deactivate your account. If you wish, you can disable this feature anytime after March 16, 2026, in the **Meta Business Suite** > **Business settings** > **Business info** panel. If you disable your contact book, it will stop storing user information, and any existing user information it has already stored will be deleted. If you re-enable the contact book later, it will start storing user information again, but previously stored information cannot be restored.

Limitations:

If you are using [Local Storage](#) and a user shares their phone number with you by tapping the share contact information button, Meta extracts the user's phone number from the shared contact card (vCard) and stores it in your contact book on Meta data centers. Only the phone number is

extracted and stored; no other vCard data is retained beyond the [standard data-at-rest period for local storage](#). Contact books are scoped to business portfolios. This means that if you have linked portfolios, a user's phone number and BSUID would have to be recorded to each portfolio's contact book independently; user contact information is not shared or synced across linked portfolios.

## Country codes

If a WhatsApp user enables the username feature, their phone number (and thus, country dialing code) may not appear in webhooks. In these cases, the user's BSUID will appear instead, prefixed with the user's [ISO 3166 alpha-2](#) two-letter country code (e.g., `US.13491208655302741918`).

## Business usernames

Businesses will also be able to adopt a business username. If you adopt a business username, however, it will not cause your business phone number to be hidden in the WhatsApp or WhatsApp Business client.

A business username is mapped to a single business phone number across all of WhatsApp, i.e. a phone number can have only one username at a given time, and no two WhatsApp phone numbers (consumer or business) can have the same username.

Business usernames must adhere to the following format:

may only contain English letters (a-z), digits (0-9), period (.) and underscore (\_) characters  
non-English characters (such as ñ, é, ü) are not supported and will cause the request to fail  
must be between 3-35 characters in length  
must contain at least one English letter (a-z, A-Z)  
must not start or end with a period or have 2 consecutive periods  
must not start with www  
must not end with a domain (e.g., .com, .org, .net, .int, .edu, .gov, .mil, .us, .in, .html, and so on)  
case is ignored when comparing usernames, but period and underscore characters are not; for example, myID and myid are the same username but myid, my.id, and my\_id are all distinct

## Reserved usernames

Before the username feature is made available, you will have the option to claim a username that WhatsApp has reserved for you. Alternatively, you can adopt a different username that aligns with your branding requirements. A reserved username can be claimed through WhatsApp Manager, Meta Business Suite, or [via API](#). Claimed usernames that are approved will become active once the username feature is made available.

If a reserved username is already in use with your Facebook Page or Instagram account, you must link your business phone number to your Facebook Page or Instagram account before you will be able to claim the username.

You can link your phone number when claiming the username in Meta Business Suite or WhatsApp Manager, or by accessing your Facebook Page or Instagram account and [adding your phone number directly](#).

To link your phone number, you must have full control of the page or account, or basic partial access with the `manage_phone` permission. See [About business portfolio and business asset permissions](#) for information about control/access and permissions.

## Chat window display priority

The following priority will be followed (in decreasing order of priority) for displaying business profile information in chat windows in the app. Your business phone numbers will always appear in your business profile.

Saved contact name  
Verified business name or [Official Business Account](#) name  
Username  
Phone number

## Support

You can contact your Partner Manager with any concerns. You can reach out to any of the [standard support channels](#); for API integrations, please raise a Direct support ticket with question type, **WA Usernames API Integration**. Use the **Report Abuse** channel via [Direct Support](#) to report impersonation. Use our [WhatsApp Intellectual Property Contact Form](#) form to report infringement.

## Get current username

`username` — Current username. Will be omitted if the business phone number has no username.  
`status` — Username status. Values can be: `approved` — The username is approved and visible to WhatsApp users. `reserved` — The username is reserved for the business phone number but is not visible to WhatsApp users. It will become visible once the usernames feature is made available to everyone.

## Delete a username

`success` — Boolean. Will be set to `true` if the username is deleted successfully, otherwise it will be set to `false`.

## business\_username\_updates webhook

A new **business\_username\_update** webhook will be added. This webhook will be triggered when a business username status changes.

Please subscribe each of your apps to this webhook field to be notified of username changes.

`id` — WhatsApp Business Account ID. `time` — Unix timestamp indicated when the webhook was triggered. `display_phone_number` — The business phone number's display number (the number displayed on your profile in the app). `username` — The username for which the status has changed. Omitted if `status` is set to `deleted`. `status` — Values can be: `approved` — Indicates the username is approved and visible to WhatsApp users. Triggered when the username's status changes from `reserved` to `approved`, or the username was changed via the WhatsApp Business app. `deleted` — Indicates the username has been deleted via the WhatsApp Business app. `reserved` — Indicates the username is reserved for the business phone number but is not visible to WhatsApp users. It will become visible once the usernames feature is made available to everyone.

## Messages

## Error codes

Adding new error code response to the [POST /<BUSINESS\\_PHONE\\_NUMBER\\_ID>/messages](#) endpoint.

Error code — 131062 Details — Business-scoped User ID (BSUID) recipients are not supported for this message.

## Webhook testing

You can test webhook payloads that reflect real-world username adoption scenarios using the **App Dashboard > Use cases** (pencil icon) > **Connect with customers through WhatsApp > Customize > Configuration** panel (**App Dashboard > WhatsApp > Configuration** for apps created before December, 2025). Click the **Test** link alongside the messages webhook to send a test messages webhook to your webhook endpoint.

The test tool supports incoming messages webhooks and status messages webhooks for sent messages, with the following scenarios:

**User has not adopted a username** — Webhook payloads will include BSUID fields and phone number fields, but no username. This represents the default state for most users at launch. **User has adopted a username and phone number is unavailable** — Webhook payloads will include the username and BSUID fields, but phone number fields will be omitted. Your integration should handle this scenario gracefully. See [Phone numbers](#) for conditions under which phone numbers are included. **User has adopted a username and phone number is available** — Webhook payloads will include all fields: username, BSUID, and phone number. **Parent BSUID present** — For businesses with linked business portfolios, webhook payloads will include a [parent BSUID](#) in addition to the portfolio-level BSUID.

## Webhook identifier quick reference

The following tables summarize which user identifiers will be included in messages webhooks, based on the type of webhook and whether the user has adopted a username.

### Outbound message status webhooks

These identifiers apply to sent, delivered, and read [status messages](#) webhooks.

Identifier	Sent to phone number	Sent to BSUID
<code>wa_id</code>	Always included	Included if phone number is available per <a href="#">Phone numbers</a> conditions
<code>user_id</code>	Always included	Always included
<code>parent_user_id</code>	Included if parent BSUIDs enabled	Included if parent BSUIDs enabled
<code>username</code>	Included in delivered/read if user has a username	Included in delivered/read if user has a username

# Incoming messages webhooks

These identifiers apply to [incoming messages](#) webhooks, including user-initiated messages and user replies.

Identifier	User has a username	User does not have a username
<code>wa_id</code>	Included if phone number is available per <a href="#">Phone numbers</a> conditions	Always included
<code>user_id</code>	Always included	Always included
<code>parent_user_id</code>	Included if parent BSUIDs enabled	Included if parent BSUIDs enabled
<code>username</code>	Always included	Not included

# Messages webhooks

## Status messages webhooks

These changes will apply to sent, delivered, read, and failed [status messages](#) webhooks.

`contacts` — New array. Only included for sent, delivered, and read status messages. Will be omitted entirely for `failed` status messages webhooks. `name` — New property. Value will be set to the WhatsApp user's display name. `username` — New property. Will be set to the WhatsApp user's username if the user has enabled the usernames feature. Will be omitted entirely for `sent` status messages webhooks, or if the user has not enabled the usernames feature. `wa_id` — New property. Will be set to the user's phone number if the phone number can be included based on the conditions described in the [Phone numbers](#) section. Will be omitted if the phone number cannot be included based on those conditions. `user_id` — New property. Will be set to the WhatsApp user's BSUID. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#) if you have enabled parent BSUIDs. Otherwise, the property will be omitted entirely. `statuses` `recipient_id` — New behavior (can be omitted). Will be set to the user's phone number, if you sent the message to the user's phone number. Will be set to the group ID, if you sent the message to a group. Will be omitted if you sent the message to the user's BSUID or parent BSUID and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section.

`recipient_user_id` — New property. Will be set to the user's BSUID or parent BSUID, if you sent the message to the user's BSUID or parent BSUID. Otherwise, it will be omitted. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#) if you have enabled parent BSUIDs. Otherwise, it will be omitted entirely.

Example delivered status messages webhook describing a message sent from a business that has enabled parent BSUIDS to the phone number of a WhatsApp user who has enabled the usernames feature:

Example delivered status messages webhook describing a message sent from a business that has enabled parent BSUIDS, to the BSUID of a WhatsApp user who has enabled the username feature. In this example, we are unable to include their phone number based on the conditions described in

the [Phone numbers](#) section (so `wa_id` and `recipient_id` are omitted).

## Incoming messages webhooks

These changes apply to incoming messages webhooks ([text](#), [image](#), [interactive](#), and so on), including incoming messages sent by users in a Group chat.

The example syntax below is for an incoming **text** message, but the changes are the same for all incoming message types.

`contacts` `profile` `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Will be omitted if the user has not adopted a username. `wa_id` — New behavior (can be omitted). Will be set to the user's phone number if the phone number can be included based on the conditions described in the [Phone numbers](#) section. Will be omitted if the phone number cannot be included based on those conditions. `user_id` — New property, set to the user's BSUID. `parent_user_id` — New property. Will be set to the user's parent BSUID, if you have enabled parent BSUIDs. Otherwise, it will be omitted. `messages` `from` — New behavior (can be omitted). Will be set to the user's phone number if the phone number can be included based on the conditions described in the [Phone numbers](#) section. Will be omitted if the phone number cannot be included based on those conditions. `from_user_id` — New property, set to the user's BSUID. `from_parent_user_id` — New property, set to the user's parent BSUID, if you have enabled parent BSUIDs. Otherwise, it will be omitted.

Example incoming text message from a user who has enabled the username feature, to a business that has enabled [parent BSUIDs](#). In this scenario, we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section.

## System status messages webhooks

These changes apply to [system status](#) messages webhooks.

`system` `body` — New string. Will be set to `User <WHATSAPP_USER_PROFILE_NAME> changed from <OLD_BSUID> to <NEW_BSUID>` if the user changed their business phone number. `wa_id` — New behavior (can be omitted). Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Will be set to the user's phone number if the user has not enabled the usernames feature. `user_id` — New property. Will be set to the user's new BSUID. `parent_user_id` — New property. Will be set to the user's new [parent BSUID](#), if you have enabled parent BSUIDs. Otherwise, it will be omitted. `type` — New value (`user_changed_user_id`). Will be set to `user_changed_user_id` if the WhatsApp user changed their phone number.

## user\_preferences webhooks

These changes will apply to [user\\_preferences](#) webhooks.

`contacts` `profile` `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Property omitted if the user has disabled the username feature. `wa_id` — New behavior (can be omitted). Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Will be set to the user's phone number if the user has not enabled the

usernames feature. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#), if you have enabled parent BSUIDs. Otherwise, it will be omitted. `user_preferences` `wa_id` — New behavior (can be omitted). Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property,. Will be set to the user's BSUID. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#), if you have enabled parent BSUIDs. Otherwise, it will be omitted.

## user\_id\_update webhooks

A new **user\_id\_update** webhook will be triggered when a WhatsApp user's BSUID changes. Subscribe your apps to this webhook field to be notified of BSUID changes.

`contacts` `wa_id` — Will be set to the user's phone number if available. Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. `user_id_update` `wa_id` — Will be set to the user's phone number if available. Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. `detail` — A human-readable description of the update. `user_id` — Object containing the user's previous and current BSUID. `previous` — The user's old BSUID. `current` — The user's new BSUID. `parent_user_id` — Object containing the user's previous and current [parent BSUID](#), if you have enabled parent BSUIDs. Otherwise, it will be omitted. `previous` — The user's old parent BSUID. `current` — The user's new parent BSUID. `timestamp` — Unix timestamp indicating when the webhook was sent.

# Groups API webhooks

## Status messages webhooks for groups

These changes will apply to `delivered` and `read` [status messages](#) webhooks for messages sent to a group.

`contacts` — New array. Only included for delivered and read status messages. Will be omitted entirely for failed status messages webhooks. `name` — New property. Value will be set to the WhatsApp user's display name. `username` — New property. Will be set to the WhatsApp user's username if the user has adopted a username. Will be omitted for sent status messages webhooks, or if the user has not enabled the usernames feature. `wa_id` — New property. Will be omitted if the user has adopted a username and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Will be set to the user's phone number, if you sent the message to the user's phone number. `user_id` — New property. Will be set to the WhatsApp user's BSUID. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#) if you have enabled parent BSUIDs. Otherwise, it will be omitted. `recipient_participant_id` — Changed. Will be set to the user's phone number, if the message was sent to their phone number. Otherwise, it will be omitted. `recipient_participant_user_id` — Will be set to the user's BSUID or

parent BSUID, if you sent the message to the user's BSUID or parent BSUID. Otherwise, it will be omitted. `recipient_participant_parent_user_id` — New property. Will be set to the user's parent BSUID if you have enabled parent BSUIDs. Otherwise, it will be omitted.

## group\_participants\_update webhooks

These changes apply to the [group\\_participants\\_update](#) webhook.

`input` — New value (BSUID or parent BSUID). Will be set to the user's phone number if you removed the user from the group using their phone number. Will be set to the user's BSUID or parent BSUID if you removed the user from the group using their BSUID or parent BSUID. `wa_id` — New behavior (can be omitted). Will be omitted if the user has enabled the username feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — New property. Will be set to the user's [parent BSUID](#) if you have enabled parent BSUIDs. Otherwise, it will be omitted. `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted.

# Coexistence

## History webhooks

These changes will apply to [history](#) webhooks that describe an onboarded business customer's WhatsApp Business app chat history.

`id` — New behavior (can be omitted). Will be omitted if, at the time of the history sync request, the user has already enabled usernames and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `context` — New context object. `wa_id` — New property. Will be omitted if, at the time of the sync request, the user has already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted. `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted. `messages` `from` — New behavior (can be omitted). Will be omitted if, at the time of the sync request, the user has already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `from_user_id` — New property. Will be set to the user's BSUID. `from_parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted.

These changes will apply to [history](#) webhooks that describe a media asset sent from a WhatsApp user to a business customer, or vice-versa.

`contacts` — New object. `profile` `username` — New property. Will be set to the user's username, if

the user has enabled the username feature. Otherwise, it will be omitted. `wa_id` — New property. Will be omitted if, at the time of the sync request, the user has already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted. `messages` `from` — New behavior (can be omitted). Will be omitted if, at the time of the sync request, the user has already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `from_user_id` — New property. Will be set to the user's BSUID. `from_parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted. `message_echoes` `to` — New behavior (can be omitted). Will be omitted if, at the time of the sync request, the user has already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `to_user_id` — New property. Will be set to the user's BSUID. `to_parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted.

## smb\_message\_echoes webhooks

These changes will apply to [smb\\_message\\_echoes](#) webhooks.

`contacts` — New array. `profile` `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted. `wa_id` — New property. Will be omitted if, at the time when the business customer used the WhatsApp Business app to send the message to the user, the user had already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted. `message_echoes` `to` — New behavior (can be omitted). Will be omitted if, at the time when the business customer used the WhatsApp Business app to send the message to the user, the user had already enabled the usernames feature and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `to_user_id` — New property. Will be set to the user's BSUID. `to_parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted.

## smb\_app\_state\_sync webhooks

These changes will apply to [smb\\_app\\_state\\_sync](#) webhooks.

`phone_number` — New behavior (can be omitted). Will be omitted if, at the time of the sync request, the user has already enabled usernames and we are unable to include their phone number based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be set to the user's phone number. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#), if you enabled parent BSUIDs. Otherwise, it will be omitted.

`username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted.

## Revoke messages webhooks

These changes will apply to [revoke messages](#) webhooks.

`contacts` `profile` `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#) if you enabled parent BSUIDs. Otherwise, it will be omitted.

## Edit messages webhooks

These changes will apply to [edit messages](#) webhooks.

`contacts` `profile` `username` — New property. Will be set to the user's username, if the user has enabled the username feature. Otherwise, it will be omitted. `wa_id` — New property. Will be omitted if, at the time when WhatsApp user edits the message, the user has already enabled the usernames feature and the phone number cannot be included based on the conditions described in the [Phone numbers](#) section. Otherwise, it will be omitted. `user_id` — New property. Will be set to the user's BSUID. `parent_user_id` — Will be set to the user's [parent BSUID](#) if you enabled parent BSUIDs. Otherwise, it will be omitted.

## Analytics

No changes.

## Billing and invoicing

No changes.

## FAQs

### What do I need to do to support usernames?

BSUIDs and parent BSUIDs will begin appearing in webhooks payloads in March 2026, before usernames are made available to WhatsApp users. In order to process messages from users who enable the feature once it is available, you will need to support BSUIDs (and parent BSUIDs if you enable them). To do this, you must:

Update your webhook integrations to support BSUIDs (and [parent BSUID](#), if using). Build logic to support handling multiple identifiers (phone numbers from non-username adopters; BSUIDs from username adopters if phone number is not present in webhooks), and map relevant fields back to your CRM/database. Update internal and external systems related to these integrations to be able to handle BSUIDs and join with previous identifiers; primarily CRM (either 3P or internal database) and any tools or workflows triggered off of CRM (e.g., triggered campaign messages, campaign management, measurement, billing, and so on). If you still require customer phone numbers,

update your messaging bots/journeys (if used) to request phone numbers, handle scenarios where users do not share phone numbers, and iterate on these new conversation journeys. See [WhatsApp Business Solution Terms](#) for general restrictions in AI use cases. If you have multiple business portfolios with Meta, you may want to implement a solution to enable central CRM access across multiple portfolios, to minimize the operational overhead that comes with using and storing BSUIDs (and parent BSUIDs).

### **When will I receive a BSUID or parent BSUID vs. a phone number?**

When a user adopts a username, they will have phone number privacy meaning their phone number will not be displayed in the app, and their phone number will not be included in webhooks. If the user's phone number is not present (the wa\_id property is missing), you can use their BSUID (or [parent BSUID](#), if using), which will be included and assigned to a new user\_id property ( `parent_user_id` for parent BSUIDs).

If a user has not adopted usernames, you will receive both their phone number and their BSUID (and parent BSUID, if enabled).

Note that will continue to share the phone number if [certain conditions are met](#). Per our Cloud API Terms of Service, however, phone numbers and related data are retained for a maximum of 30 days to support features like message redelivery. There may be situations where you receive messages from existing users outside of this 30 day window, which may look like a new user thread to you. Therefore, it is essential that you begin supporting BSUIDs as soon as possible, to minimize losing any conversation context.

### **Why do partners and directly-integrated businesses using the Cloud API, including directly integrated ads that click to WhatsApp advertisers, have to adopt BSUID?**

Partners and businesses must adopt BSUID to continue processing incoming messages from WhatsApp username adopters. Once BSUID is adopted and user messages from username adopters are processed, message webhooks will no longer include phone numbers in some cases as part of the webhook such as wa\_id, so anyone using the Cloud API must ensure all connected systems can handle BSUID. They will also be able to ask for a user's phone number in-thread.

### **If I have not yet adopted BSUIDs and start to receive messages from username adopters which I cannot process, recourse is there?**

If you have not yet adopted BSUID and are not able to process messages from username adopters, there will not be any recourse or corrective action that you can take.

For messages from new customers: the webhook will continue to be sent of an incoming message. Depending on the specifics of the implementation, this may impact your systems that are not equipped to handle incoming messages without user phone numbers, and BSUID assigned to the new user\_id field. For messages from your existing customers: the phone number will continue to be included if the conditions described in the [Phone numbers](#) section are met.

Once you support BSUIDs, request phone numbers from users by implementing a [phone number request button](#).

### **How do business usernames differ from display names? When will a user see a business username vs a display name?**

Business usernames will provide an ability for the users to reach the business by the business' username, meaning an end user can search for a business username using their exact username and reach out to the businesses. Since end users cannot search by display names, business usernames offer a clear advantage as a searchable and unique identifier for users to reliably find the correct business.

Business usernames must follow specific formatting rules on length and allowed characters, while display names have some more leeway in terms of formatting.

Business usernames are unique and are tied 1-1 to phone numbers, meaning @JaspersMarket would be tied to one phone number while @JaspersMarketCustomerSupport would be tied to another phone number. Display names are not tied 1-1 to phone numbers, meaning the display name Jasper's Market can have 10 phone numbers under this display name.

When a business has both a username and display name, display name will be shown first (e.g., in Profile, Chat list, Messages, and so on.), for the businesses to build trust with the users and for users to recognize the business when business reaches out to the user.

## Document changelog

### March 31, 2026

Updated BSUID webhook rollout date from March 31 to early April 2026. Updated [contact book](#) limitations: Local Storage businesses now automatically have phone numbers extracted from shared vCards and stored in the contact book on Meta data centers. Updated [requesting phone numbers from users](#): removed the Local Storage exception that required manually sending a message; replaced with automatic vCard phone number extraction behavior. Updated [webhook testing](#) instructions with corrected App Dashboard navigation path. Added availability warning to the [adopt or change a business username](#) section.

### March 23, 2026

Corrected [send message response](#) and [send marketing message response](#) `user_id` descriptions: when both a phone number and BSUID or parent BSUID are included in the request, the response will not include `user_id`, since the phone number takes precedence and the response is identical to a phone-number-only request. Updated the example response for the phone number and BSUID case accordingly.

### March 18, 2026

Added [webhook identifier quick reference](#) tables summarizing which identifiers appear in messages webhooks. Added [webhook testing](#) section describing test scenarios available in the App Dashboard. Added [user\\_id update webhooks](#) section. Added [get blocked users](#) response changes to the Block Users API section. Added [parent BSUID](#) format description and updated example payloads to use the `ENT` prefix (e.g., `US.ENT.11815799212886844830`). Added availability note for the [request contact information button](#) (early May 2026). Clarified that the [contact book](#) is provided and hosted by Meta with no integration work required, is scoped to the business portfolio level, and only stores interactions that occur after launch. Clarified that [user usernames](#) have the same format restrictions as [business usernames](#). Clarified that non-English characters (such as ñ, é, ü) are not supported in [business usernames](#). Clarified that the 30-day lookback conditions in the [Phone](#)

[numbers](#) section are evaluated per business phone number. Removed “You are in the user’s WhatsApp contacts list” from the [phone number inclusion conditions](#).

## February 18, 2026

Clarified that API requests that accept both a phone number and a BSUID or parent BSUID can include both identifiers simultaneously, with the phone number taking precedence. Updated [send message requests](#), [send marketing message requests](#), [business-initiated call requests](#), and [block or unblock user requests](#).

## February 6, 2026

Changed number of alphanumeric characters that compose [BSUIDs](#) from 256 to 128 alphanumeric characters. Changed how to use a BSUID to send a message; BSUIDs must now be assigned to dedicated properties/fields in message send requests (instead of existing properties/fields supporting both BSUIDs and phone numbers). Changed how [country codes](#) will appear in webhooks: they will be prefixed to user BSUIDs instead of assigned to a dedicated webhook property. Added [parent BSUID](#) information, which can be used across linked business portfolios. Added [contact book](#) information, which can automatically store user phone numbers and BSUIDs. Added [phone number request button](#) information. Changed syntax examples, payload examples, and descriptions for all webhooks that returned empty strings in cases where a user has enabled the usernames feature. Now, these properties will not be set to an empty string. Instead, they will be omitted (e.g, the `wa_id` property in incoming messages webhooks). Changed how errors are returned when attempting to [adopt or change a business username](#). Removed ability to cancel pending business username requests. Changed `phone_number_username_update` webhook to [business username updates](#) webhook.