BetterImpact > Swift User Ingest

Overview

Field	Value
Workflow ID	DmoxddEkpx854rYj
n8n URL	https://n8n.pbr.org.au/workflow/DmoxddEkpx854rYj
Status	Active
Trigger	Schedule — every 24 hours
Systems	BetterImpact (source), Swift Digital (destination), n8n Data Table (audit log)
Predecessor	BetterImpact>Swift User Ingest_old (inactive, superseded)

Purpose

This workflow keeps Swift Digital contact groups in sync with BetterImpact, PBR's people management system. It runs once every 24 hours and processes all BetterImpact users whose records have been updated in the past 48 hours.

BetterImpact is treated as the source of truth. For each recently changed person, the workflow determines their engagement type (Volunteer, Staff, Board/Sub-Committee, or Guild Member) and ensures their Swift Digital contact record and group memberships reflect that classification. If a person no longer qualifies for a group they are in, they are removed. If they are not yet in a group they should be in, they are added. If they do not exist in Swift Digital at all, a new contact record is created.

Data Sources

BetterImpact API

Authenticates via HTTP Basic Auth. Two separate API calls are made at the start of each run:

- **GetAllBIUsers Updated Last 48Hrs** — fetches all users from the *PBR - Whole Team* organisation updated in the last 48 hours, paginated at 100 records per page.
- **GetAllGuildBIUsers Updated last 48Hrs** — fetches all users from the *PBR - GUILD* organisation updated in the last 72 hours (slightly wider window), paginated at 100 records per page.

Both result sets are merged and aggregated before processing. The wider 72-hour window for Guild users is intentional, accounting for the fact that Guild membership changes may be recorded slightly later in BetterImpact.

Fields Extracted from BetterImpact

For each user, the following fields are extracted and normalised:

Field	Source in BetterImpact	Notes
`BI_user_id`	`user_id`	BetterImpact's unique user identifier
`first_name`, `last_name`, `title`	Top-level fields	
`email`	`email_address`	Parsed with `extractEmail()` to strip display-name formatting
`mobile`	`cell_phone`	Spaces removed
`volunteer_status`	`memberships[0].volunteer_status`	Used to determine active vs archived status. Value must be `Accepted` to be considered active.
`engagement_type`	Custom field category *PBR VOLUNTEER TYPE*	Determines which Swift group the person belongs to
`org_name`	`memberships[0].organization_name`	Used to distinguish Guild users in the GUILD org from those appearing in Whole Team
`branch`, `sub_branch`	Custom fields *Branch* and *Sub-Branch*	Ampersands (`&`) are replaced with *and* before sending to Swift Digital, which does not accept the `&` character in these fields
`country`	Looked up via n8n Data Table mapping from `country_name`	Converted to ISO country code(s) for Swift Digital compatibility

Users without an email address are filtered out early and never sent to Swift Digital.

Engagement Type Classification

Each person is routed to one of four Swift Digital groups based on their `engagement_type` value from BetterImpact:

Engagement Type	Swift Digital Group	Match Logic
Contains *Volunteer*	PBR_Vols	String contains check
Contains *Staff*	PBR_Staff	String contains check
Equals *Board Member* or *Sub-Committee Member*	PBR_BoardSubcommittees	Exact match (OR)
Equals *PBR Guild Member* AND org is *PBR - GUILD*	PBR_Guild (AllGuild)	Both conditions must match

Each person is evaluated against all four type checks independently in parallel. A person could theoretically match more than one (e.g. if their engagement type changes mid-run), though in practice a person should match only one.

Processing Logic — Per User

Once a user's engagement type is determined, the workflow follows this decision tree for each type. The logic is identical across all four types — described here using Volunteer as the example:

Active User Path (volunteer_status = Accepted)

1. **Does the user exist in Swift Digital?** (checked via `contact_ids` — a sentinel value of `zzzz69363c11d9fa7821` indicates no match was found from the email search)
   - **Yes — user exists:** Update their contact details in Swift Digital (name, email, mobile, country, engagement type, BetterImpact user ID). Then check whether they are already a member of the correct group.
     - If **not in the group**: add them to it.
     - If **already in the group**: no group action needed.

- **No — user does not exist:** Create a new Swift Digital contact record with all their details, and assign them to the correct group in the same API call.

Inactive/Archived User Path (volunteer_status != Accepted)

1. **Does the user exist in Swift Digital?**
   - **No — user does not exist:** Nothing to do, exit.
   - **Yes — user exists:** Check whether they are a member of the group they should no longer be in.
     - **Not in the group:** Nothing to do, exit.
     - **In the group — and it is their only group:** Delete the contact from Swift Digital entirely.
     - **In the group — and they are in other groups too:** Remove them from this specific group only (do not delete the contact).

The "last group" check (delete vs group-remove) is done by checking whether `group_ids.length <= 1`. If a contact is only in one group and should be removed from it, deleting the contact entirely is the correct action.

Guild User Edge Case

Guild members appear in two BetterImpact API results: the *PBR - Whole Team* response and the *PBR - GUILD* response. To prevent a Guild member from being incorrectly processed as a Whole Team volunteer, the workflow uses a deduplication merge that prefers the *PBR - Whole Team* record when a user appears in both. The `Is Guild?` check then additionally requires that `org_name = PBR - GUILD`, ensuring only genuine Guild organisation records are routed to the Guild group.

There is also a specific filter called *Catch Guild User Archived in BI Whole Team*. This handles a known data pattern in BetterImpact where a Guild member who has been archived in the GUILD organisation still appears in the Whole Team result. This filter passes the user through to the Guild group removal logic, preventing them from being incorrectly left in the Swift Digital Guild group after archival.

Duplicate Contact Detection

After main processing, the workflow runs a secondary check for duplicate Swift Digital contacts. For each processed user, it searches Swift Digital by BetterImpact User ID (`Internal_BIUserId`) to see if more than one contact record is associated with that ID. If duplicates are found, the workflow:

1. Retrieves full details for all duplicate contacts.
2. Identifies the earliest-created contact by `create_stamp`.
3. Deletes the earliest contact, keeping the most recently created one.

The premise is that BetterImpact is the source of truth — if a single BI user ID maps to multiple Swift contacts, the oldest is assumed to be stale and the newest is retained.

---

n8n Data Table (Audit Log)

After processing, each user record is upserted into an n8n internal Data Table named **BI_Users** (project ID: `JMezsOufXlA5w9MB`, table ID: `ps5tTQfbPJ8IMtfQ`). This provides an internal record of the last-known state of each processed user including their Swift contact ID, group IDs, engagement type, and email. The upsert matches on `BI_user_id`.

This table is used as a debugging and audit reference — it is not consumed by any downstream automated workflow.

Custom Fields Written to Swift Digital

When creating or updating a contact in Swift Digital, the following custom internal fields are populated:

Swift Field	Value
`Internal_EngagementType`	The person's engagement type from BetterImpact
`Internal_BIUserid`	The person's numeric BetterImpact user ID
`Internal_Branch`	Branch (omitted if null)
`Internal_SubBranch`	Sub-branch (omitted if null)

Note: Branch and Sub-Branch fields were present in an earlier version of the workflow but are not written in the current active version's update/create payloads. They remain in the BetterImpact data extraction step.

API Rate Limiting

Swift Digital API calls are batched at 100 requests per batch with a 7,500ms interval between batches for search and group-read operations. Contact creation for Volunteers uses a 5,000ms interval. This prevents hitting Swift Digital API rate limits during large sync runs.

Credentials

Credential	Type	Used For
BetterImpact API	HTTP Basic Auth	All BetterImpact API calls
Swift Digital OAuth2	OAuth2	All Swift Digital API calls

Maintenance Notes

- **Schedule:** Runs every 24 hours. The 48-hour lookback window means a missed run will still catch changes from the previous cycle on the next run.
- **BetterImpact engagement type values:** If PBR adds new engagement types in BetterImpact, the *Is Vol?*, *Is Staff?*, *Is Board Or Sub-Committee?*, and *Is Guild?* nodes must be updated to include the new values, and a corresponding Swift group and routing branch must be added.
- **Swift Digital group IDs** are stored in the n8n Global Constants node (not hardcoded in individual nodes). If group IDs change in Swift Digital, update the Global Constants node only.
- **Country code mapping** is maintained in the n8n Data Table referenced by ID `wRiJ3eMdq66p8Efh`. If new countries appear in BetterImpact, add a mapping row to that table.
- **The sentinel value** `zzzz69363c11d9fa7821` is used as a placeholder to indicate no Swift contact was found for a given email. This is an internal n8n pattern — do not remove it from the contact_ids field checks.
- **The _old workflow** (`0FJoBOuSCoGadbOY`) is inactive and retained for reference only. Do not activate it.

Revision #1
Created 2026-05-07 06:15:37 UTC by PBR_Documentation
Updated 2026-05-07 06:15:37 UTC by PBR_Documentation