

DEXPRO AWF External Approvals — SharePoint Integration Contract

This document describes the **SharePoint list contract** between Business Central (the AWF extension) and any external system that consumes or responds to approval requests. It is **transport-neutral**: the reference implementation uses Power Automate + Microsoft Teams (see `SETUP-GUIDE.md`), but any system that can read and write SharePoint list items via the Microsoft Graph API — or the native SharePoint REST API — can participate.

“ **Audience:** integration developers. End-user setup is covered in `SETUP-GUIDE.md` .

1. Design principles

1. **The SharePoint list is the queue.** BC writes a list item when an external approver must decide; the consumer writes back to the same item with the decision; BC's Job Queue poller picks it up.
2. **BC is the system of record.** The SP item is transient — BC deletes it after processing. Consumers must not rely on SP history.
3. **Stable internal column names.** The column `name` (not `displayName`) is the contract. Display names are localized and must not be used.
4. **Forward compatibility.** BC writes a `SchemaVersion` value on every item. Consumers should reject / log items with an unsupported version instead of silently misbehaving.
5. **Idempotency.** BC only cares about the first terminal status a consumer writes. Writing the same status twice is a no-op.

2. Polling model

- BC runs a Job Queue entry (`Codeunit 70954832 "DXP AWF Ext. Approval Poller"`) on a recurring interval (default: every 2 minutes).
- The poller fetches items that match `Processed eq false and ApprovalStatus ne 'Pending' and BCCompanyName eq '<current company>'` (server-side filter; falls back to client-side if

Graph declines the filter). The company clause is always applied — see § 7a.

- For each matched item, BC updates the corresponding AWF entry, calls the workflow engine, then sets `Processed = true` on the SharePoint item and attempts to `DELETE` it.
- If `DELETE` fails, the `Processed = true` flag is enough to exclude the item from subsequent polls; orphaned items are retried every cycle by `CleanupOrphanedSPItems`.

Implication for consumers: you have between 0 and N minutes (where N is the polling interval) between writing the decision and BC processing it. Do not expect synchronous confirmation. If you need a confirmation back, watch for the item's `Processed` field to flip or the item to disappear.

3. Column reference

Every column below uses **the internal** `name` — that is the contract. Display names may be localized.

3.1 Inputs (BC ? consumer)

BC populates these when the item is created. Consumers **must not modify** them.

Name	Type	Notes
<code>SchemaVersion</code>	Text	Currently <code>"1"</code> . Bump indicates a breaking change in this contract.
<code>BCCompanyName</code>	Text	The BC company that owns the entry. Always set on new rows; must not be changed by consumers. BC's per-company poller uses this to ignore items from other companies (see § 7a).
<code>Title</code>	Text	Short record description (e.g. purchase order no.). Part of the built-in SharePoint column.
<code>BCEntryNo</code>	Text	Entry No. of the AWF <code>Ext. Approval Entry</code> record within <code>BCCompanyName</code> . Not globally unique — always pair with <code>BCCompanyName</code> when correlating.
<code>BCApprovalEntryNo</code>	Text	Entry No. of the underlying BC standard <code>Approval Entry</code> (table 454). Identical across all peer rows in a group — useful for grouping rows by their underlying decision unit (e.g. cancelling peers when one claims).
<code>BCInstanceID</code>	Text	AWF workflow instance GUID — globally unique across companies.

Name	Type	Notes
Description	Text	Localized RecordId text of the approved record (<code>Format(RecId, 0, 1)</code> , e.g. <i>Purchase Line: Invoice,EKRECH1052,20000</i> with translated table/field captions). Shown on the card as the record line; works for any table.
ApproverEmail	Text	UPN / mail address. Route notifications to this address.
ApproverDisplayName	Text	Human-readable name.
AttachmentsUrl	Text	SharePoint sharing link to a folder containing document attachments. Empty when upload is disabled or no attachments exist.
BCDocumentUrl	Text	Direct link to the source record in Business Central. Opens for approvers whose Business Central licence grants web-client access; others see a sign-in wall. Empty if BC could not derive a card-page URL for the record.
RecordDetails	Multi-line text	CommonMark markdown rendering of the approved record's own fields (e.g. for a Purchase Line: Type, No., Description, Quantity, Direct Unit Cost, Line Amount). Renderable directly in an Adaptive Card <code>TextBlock</code> with <code>wrap=true</code> . Curated for the common Sales / Purchase header & line tables; falls back to a generic field walk for other tables.
IsGroupApproval	Boolean	<code>true</code> when this item is one of several for a claim-based group approval.
GroupCode	Text	External approver group code (only set when <code>IsGroupApproval = true</code>).
Amount	Text	Display-formatted amount (locale-dependent).
AmountRaw	Number	Machine-readable amount (always point-decimal). Prefer this over <code>Amount</code> .
DueDate	Text	Display-formatted date.
DueDateRaw	DateTime	ISO 8601, UTC. Prefer this over <code>DueDate</code> .
CreatedRaw	DateTime	ISO 8601, UTC. When the item was created by BC.

Name	Type	Notes
<code>SenderName</code>	Text	Display name of the requester (<code>User."Full Name"</code>); falls back to the BC user id when no full name is recorded.
<code>StageName</code>	Text	Workflow stage description.
<code>LblTitle</code> ... <code>LblClaimedSuccessMsg</code>	Text	Pre-translated UI strings and message templates for the approver's language (see 3.4). The <code>LblAlready...Msg</code> and <code>LblClaimedSuccessMsg</code> columns carry message templates with <code>{title}</code> / <code>{claimer}</code> placeholders that the consumer (e.g. Power Automate) substitutes at runtime.

3.2 Response fields (consumer ? BC)

The consumer writes these once a decision is made. Only `ApprovalStatus` is required.

Name	Type	Required	Notes
<code>ApprovalStatus</code>	Choice	yes	One of <code>Approved</code> , <code>Rejected</code> , <code>Claimed</code> . Case-insensitive on the BC side. SharePoint presents a dropdown with all valid values; the initial/default value is <code>Pending</code> . Any non-terminal value is logged by BC and the item is left alone.
<code>ResponseComment</code>	Multi-line Text	no	Free text, max 2048 chars on the BC side. SharePoint renders a multi-line text area for editing. Shown in BC's approval comment trail.
<code>ResponseDateTime</code>	DateTime	no	ISO 8601 UTC via Graph, native picker in SP. If omitted, BC uses the current time.

Name	Type	Required	Notes
ClaimedBy	Text	yes when ApprovalStatus = Claimed	Email of the group member who claimed the entry. BC also writes this column on peer items that get cancelled because someone else in the group claimed first — in that case it identifies the user who took the entry over (the SP item's ApprovalStatus will simultaneously be Cancelled).
TeamsMessageId	Text	no	The Microsoft Teams message ID of the adaptive card the consumer posted for this item. A consumer that wants to close peer cards (replace a group member's still-open card once another member decides) writes this back right after posting the card, and reads peers' values during the claim cascade to target <i>Update an adaptive card</i> . BC also stores it (and accepts it via the setTeamsMessageId bound action) for diagnostics. Optional — leave empty if the consumer doesn't implement card-closing.

“ **Do not write** Processed, SchemaVersion, BCEntryNo, BCInstanceID, or any Lbl* column. BC owns them. (TeamsMessageId is the one consumer-writable transport column besides the response fields above.)

3.3 BC-managed state

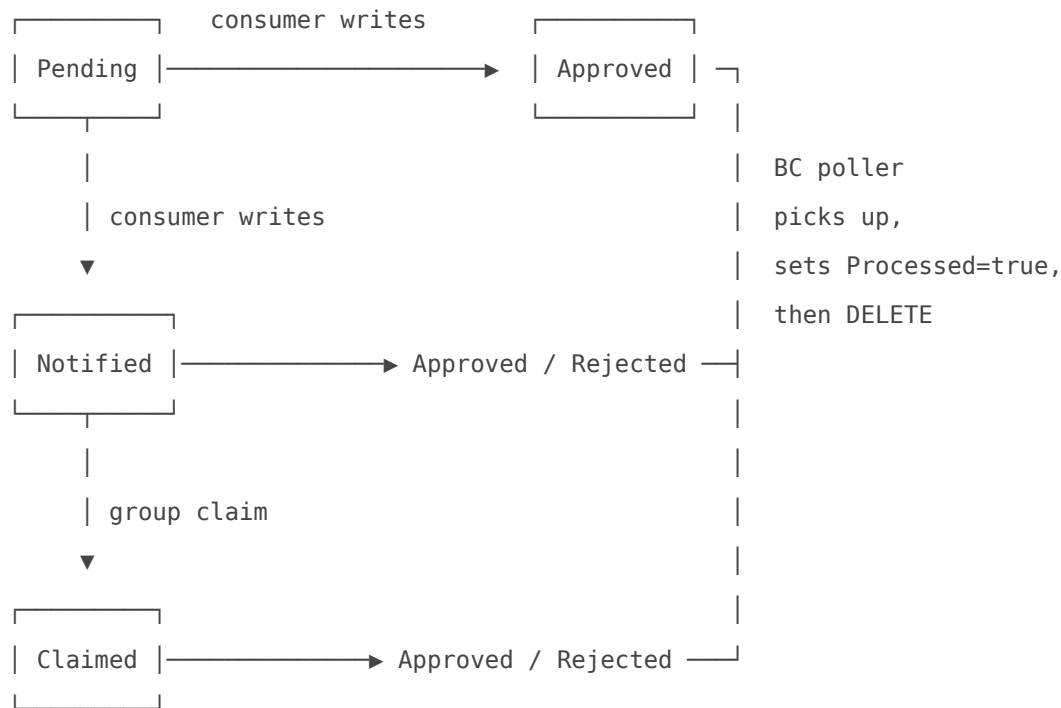
Name	Type	Written by	Purpose
Processed	Boolean	BC only	true once BC has consumed the response. Poller excludes these from subsequent fetches. Consumers must treat as read-only.

3.4 Localized label columns

BC pre-translates the adaptive-card / UI labels for the approver's language and writes them as plain text. Consumers can use them directly without calling BC or a translation service.

Name	Typical English value
LblTitle	Approval Request
LblApprove	Approve
LblReject	Reject
LblComment	Comment
LblClaim	Claim
LblRelease	Release
LblAmount	Amount
LblSender	Requested by
LblDueDate	Due Date
LblStage	Stage
LblDescription	Description
LblAttachments	Approval Documents
LblOpenInBC	Open in Business Central
LblDetails	Details
LblAlreadyClaimedByMsg	This approval ({title}) has already been claimed by {claimer}. You can safely ignore this card.
LblAlreadyHandledMsg	This approval ({title}) has already been handled by another approver or cancelled in Business Central. You can safely ignore the earlier card.
LblClaimedSuccessMsg (display name <i>Decision Confirmation</i>)	The approval entry "{title}" has been successfully processed.
LblApprovedResult (display name <i>Approved (Result)</i>)	Approved
LblRejectedResult (display name <i>Rejected (Result)</i>)	Rejected
LblResultTitle (display name <i>Result Title</i>)	Approval Decision
LblClosedTitle (display name <i>Closed Title</i>)	No Longer Open
LblResponseSent (display name <i>Response Sent</i>)	Response sent.

4. ApprovalStatus state machine



- **Pending** — written by BC on creation.
- **Notified** (*optional*) — consumer can set this after the approver has been notified but before they respond. Purely informational; BC does not act on it.
- **Claimed** — group approvals only. Consumer must also set **ClaimedBy**.
- **Approved / Rejected** — terminal. Consumer may also set **ResponseComment** and **ResponseDateTime**.
- **Processed**, **Error**, **Cancelled** — BC-internal terminal states. Do not write them.

4a. Approving directly in SharePoint (no consumer app)

Because the list uses typed columns, a human can approve an entry by opening the list item in SharePoint and editing it — no Teams, no Power Automate, no BC access required:

1. Open the SharePoint approval list in a browser.
2. Click the pending item. SharePoint opens the edit form.
3. Pick a terminal value from the **Approval Status** dropdown: **Approved**, **Rejected**, or (for group approvals) **Claimed**.
4. Optionally type a reason in the **Response Comment** text area.
5. Optionally set **Response DateTime** (left blank, BC uses the current time).
6. Click **Save**.

BC's Job Queue poller picks up the change on the next cycle (default: every 2 minutes) and processes it through the approval workflow identically to a Power Automate write-back.

Permissions: the SharePoint user needs Edit permission on the list. Because the `Processed`, `SchemaVersion`, `BCEntryNo`, and `Lbl*` columns are BC-owned, consider using a SharePoint view or column-level formatting to hide them from the edit form — users only need `ApprovalStatus`, `ResponseComment`, `ResponseDateTime`, and (for claim) `ClaimedBy`.

5. Minimal response payload

Approve

```
PATCH /sites/{siteId}/lists/{listId}/items/{itemId}/fields HTTP/1.1
Authorization: Bearer ...
Content-Type: application/json

{
  "ApprovalStatus": "Approved",
  "ResponseComment": "Looks good to me.",
  "ResponseDateTime": "2026-04-22T10:15:00Z"
}
```

Reject

```
{ "ApprovalStatus": "Rejected", "ResponseComment": "Duplicate PO." }
```

Claim (group approval)

```
{ "ApprovalStatus": "Claimed", "ClaimedBy": "maria@contoso.com" }
```

6. Alternative: native BC API

The SharePoint list is the recommended integration point because the approver works entirely in Teams/SharePoint rather than the BC web client. Note that acting on BC data still requires each approver to hold an appropriate **paid Business Central user licence** — routing responses through a service account does **not** remove the per-user licence requirement (Microsoft

multiplexing / indirect-access terms). DEXPRO makes no licensing representation; the customer and their Microsoft licensing partner are responsible for confirming licensing with Microsoft. If your system has a BC service-to-service principal, you can bypass SharePoint entirely and post directly to BC:

- **Publisher:** `dexpro`
- **Group:** `advancedWorkflow`
- **Version:** `v1.0`
- **Entity:** `externalApprovalEntries`
- **Bound actions:** `approve`, `reject`, `claim`, `releaseClaim`, `markNotified`

Example:

```
POST
/api/dexpro/advancedWorkflow/v1.0/companies({id})/externalApprovalEntries({entryNo})/Microsoft
.NAV.approve
{ "comment": "Looks good to me." }
```

Both paths — SharePoint write-back and the BC API — converge on the same `ProcessApprovalResponse` code path and are fully interchangeable.

7a. Multi-company BC installs

A single SharePoint list can serve multiple BC companies. Each company's job-queue poller runs independently in its own company context, and the poller's server-side `$filter` includes `fields/BCCompanyName eq '<current company>'` so a company only picks up its own items. There is a belt-and-braces client-side re-check on every page — rows whose `BCCompanyName` does not match the current company are discarded even if the server-side filter was bypassed.

Implications for consumers:

- Always treat `(BCCompanyName, BCEntryNo)` as the composite key. `BCEntryNo` alone is only unique within a single company.
 - Do not write or modify `BCCompanyName`. BC sets it on item creation and never changes it.
 - `BCInstanceID` (GUID) is globally unique and is a safe correlation key without `BCCompanyName`.
 - If you build a Power Automate flow that writes items (e.g. a custom intake), set `BCCompanyName` to the target company or BC's poller will never see your items.
-

7. Versioning

- **Additive changes** (new columns, new optional response fields) will **not** bump `SchemaVersion`. Consumers must ignore unknown fields.
- **Breaking changes** (renamed columns, changed semantics, required response fields) will bump `SchemaVersion`. BC will continue to write the old schema version until a deprecation release is announced.
- Consumers should validate `SchemaVersion` on every item and refuse items whose version they do not understand.

8. Troubleshooting checklist for integrators

Symptom	Likely cause
BC never processes a response	<code>ApprovalStatus</code> not one of <code>Approved</code> / <code>Rejected</code> / <code>Claimed</code> ; or consumer wrote <code>Processed=true</code> ; or AWF Setup has <code>SP Approvals Enabled = No</code> .
Item keeps reappearing in polls	<code>Processed</code> was not touched by the consumer (expected) and DELETE fails server-side — check Graph app permissions (<code>Sites.ReadWrite.All</code>). BC marks <code>Processed=true</code> before delete, so a single <code>Processed</code> flip in the UI is typically not needed.
Claim succeeds for two members	Consumer wrote <code>ApprovalStatus=Claimed</code> without checking the current status first. Prefer the BC API <code>claim</code> bound action — it enforces optimistic locking.

Revision #2
Created 2026-06-24 12:18:47 UTC by Bernd Feddersen
Updated 2026-06-24 12:18:52 UTC by Bernd Feddersen