**Step‑by‑Step Build Guide for the ROI Newsletter Agent (“ROI‑Pilot”).**

Follow it top‑to‑bottom to go from zero to “**Draft newsletter**” → **HTML ready to paste in Outlook**.

\*Notes: This can be done for any team, just change ROI for your team’s name

**0) What you’ll build**

An agent that, when you type **Draft newsletter**, will: 1) Read four Excel **tables** (Engagement Summary, Customer Highlights, ROI Team Highlights (individual), Kudos) from **SharePoint**.

2) Read last week’s meeting notes from **OneNote** in SharePoint.

3) Write a friendly, enthusiastic, positive **HTML newsletter** in a predefined structure, ready to **copy & paste** into Outlook.

Phase 1 is **human‑in‑the‑loop**: the agent **does not send** the email. You’ll paste the HTML into Outlook and send it.

**1) Prerequisites checklist**

* Copilot Studio (maker access to your environment).
* Power Automate (cloud flows).
* SharePoint site where the Excel file and OneNote live.
* Permissions to read/write the SharePoint library and OneNote section.
* Browser pop‑ups allowed for make.powerautomate.com (needed for connector sign‑ins).

**2) Create the Excel workbook (SharePoint)**

**Path suggestion:**

/Shared Documents/Newsletter/ROI\_Newsletter\_Template.xlsx

Create **four sheets**, and ensure each data range is an **Excel Table** with the **exact** Display Name:

1. **Sheet 1 — Engagement summary** → **TblEngagementSummary**

Columns: Category | Count | Notes

Rows:

* + Active Engagements
  + Completed Engagements
  + Engagements Won
  + Pending Engagements
* **Sheet 2 — Customer Highlights** → **TblCustomerHighlights**

Columns: Customer name | Licences total | Deliverable | Team member

1. **Sheet 3 — ROI teams Highlights** → **TblROIHighlights**

Columns: Name | Highlights

(Prefill names if helpful: Fernando, Stephanie, Luz, Jordan, Keith Brian, Jennifer)

1. **Sheet 4 — Kudos** → **TblKudos**

Columns: From | To | Message

**Important:** The Excel connector reads **Tables**, not loose ranges. In Excel Online, select your data → **Format as Table** → **Table Design** → set Display Name exactly as above.

**3) Prepare OneNote (SharePoint)**

* Notebook section: **ROI Team Weekly**.
* Each week, add/update the page with meeting notes (e.g., “ROI Weekly — 2025‑09‑12”).
* We’ll fetch the **latest page** and pass its text to the agent for “Most important update” and “Team highlights”.

**4) Build the Power Automate Agent Flow (tool) — “Read‑Excel‑and‑Notes”**

You’ll create a flow the agent will call to fetch deterministic data.

**4.1 Create the flow**

1. In **Copilot Studio** → open your agent (or create later in step 5) → **Tools** → **Add a tool** → **Agent flow**.
2. In **Power Automate**, choose trigger **When an agent calls the flow**.

**4.2 Add the four Excel actions**

Add **Excel Online (Business)** → **List rows present in a table** **four times**, one per table:

* **Location**: your SharePoint site
* **Document Library**: Documents
* **File**: browse to /Newsletter/ROI\_Newsletter\_Template.xlsx
* **Table**: pick each of:
  + TblEngagementSummary
  + TblCustomerHighlights
  + TblROIHighlights
  + TblKudos

Rename the actions for clarity (ellipsis ⋯ on the action header → **Rename**):

* **EngagementSummary**
* **CustomerHighlights**
* **ROIHighlights**
* **Kudos**

(If you might exceed 256 rows, open the action’s **Settings** → **Pagination** ON.)

**4.3 Pull the latest OneNote page and convert to text**

There are several ways; use this simple pattern:

1) **OneNote (Business) → Get pages for a specific section**

* Notebook: your team notebook
* Section: **ROI Team Weekly**

(If your tenant supports filtering/sorting here, choose the **latest**; otherwise, pick the first or add a small filter step.)

2) **OneNote (Business) → Get page content**

* **Page Id**: from the previous step’s latest page.

3) **Compose** (convert HTML to simple text; resilient approach):

* Add a **Compose** action named **MeetingNotesText**.

**4.4 Respond to the agent (define outputs)**

Open **Respond to the agent** → **Add outputs** exactly like this:

* **Array** engagementSummary

coalesce(outputs('EngagementSummary')?['body/value'], createArray())

* **Array** customerHighlights

coalesce(outputs('CustomerHighlights')?['body/value'], createArray())

* **Array** roiHighlightsIndividual

coalesce(outputs('ROIHighlights')?['body/value'], createArray())

* **Array** kudos

coalesce(outputs('Kudos')?['body/value'], createArray())

* **Text** meetingNotes

Value = output of **MeetingNotesText** (Compose) or '' if you skipped text conversion.

**Publish** the flow (top right).

**5) Create and configure the agent in Copilot Studio**

**5.1 Create the agent**

* **New agent** → Name: **ROI Newsletter Agent (ROI‑Pilot)**.
* Authentication: keep **Authenticate with Microsoft** (default).

**5.2 (Optional) Add SharePoint as a knowledge source**

* Agent → **Add knowledge** → **SharePoint** → paste your SharePoint **site URL**.

*(Not required for Phase 1, but handy if you want broader context later.)*

**5.3 Attach the tool (your flow)**

* Agent → **Tools** → **Add a tool** → **Power Automate flow** → pick your published flow.
* Name the tool: **Read‑Excel‑and‑Notes**.
* Description: “Reads Excel tables and latest OneNote page; returns arrays and meeting notes.”

**5.4 System instructions (paste this text)**

Note: Avoid using angle brackets in system instructions.

*ROLE: You are ROI-Pilot, a professional journalist with a friendly, enthusiastic tone, a good sense of humor, and a mission to spread positivity. You write like a top-tier newsletter editor who makes updates engaging and easy to read.*

*WHEN USER SAYS "Draft newsletter":*

*1) Call tool Read-Excel-and-Notes to retrieve:*

*- engagementSummary: rows from TblEngagementSummary (Active, Completed, Won, Pending with counts).*

*- customerHighlights: rows from TblCustomerHighlights.*

*- roiHighlightsIndividual: rows from TblROIHighlights.*

*- kudos: rows from TblKudos.*

*- meetingNotes: text from the latest OneNote meeting page.*

*2) Rewrite all content in the voice of ROI-Pilot:*

*- Friendly, enthusiastic, slightly humorous, and always positive.*

*- Make it feel like a professional newsletter, not raw data.*

*3) Structure the newsletter as follows (HTML-only):*

*1. Introduction: Warm, upbeat greeting.*

*2. The most important update from our team: summarize from meetingNotes. If empty, say: "No major team update recorded last week—but stay tuned for more wins ahead!"*

*3. Engagement Summary: Present the four categories with counts in an HTML table with inline CSS and add one short insight.*

*4. Customer Highlights: Present as an HTML table with inline CSS (Customer, Licences, Deliverable, Team Member) plus a short positive comment.*

*5. ROI Team Highlights: Summarize themes from meetingNotes in a lively tone. If none, say so.*

*6. ROI Team Highlights (Individual): List each name and highlight (bullets or a mini-table) and add a friendly remark.*

*7. Kudos: Bullet list with a positive twist.*

*4) Output ONLY the HTML version, ready to copy and paste into Outlook (inline CSS, no external styles). If any section has no data, include a placeholder like: "No updates this week — keep the momentum!"*

*CONSTRAINTS:*

*- Never send emails. Only prepare the newsletter in chat for copy-paste.*

*- Keep tone consistent: professional journalist + friendly + positive + a dash of humor.*

*- Always make the newsletter engaging and easy to read.*

**Publish** the agent.

**6) Test end‑to‑end (your first run)**

1. **Populate Excel**:
   * MiX export → paste into **Sheet 1** table.
   * Fill **Customer Highlights** (Sheet 2), **ROI Highlights (individual)** (Sheet 3), **Kudos** (Sheet 4).
2. Ensure OneNote has last week’s notes.
3. Open the agent → type **Draft newsletter**.
4. The agent will call **Read‑Excel‑and‑Notes** and return a single **HTML** newsletter.

**Paste into Outlook**:

* If you see HTML code instead of rendering, use Outlook Desktop → **Insert → Attach File → Browse This PC** → select a saved .html file of the output → click the small arrow on **Insert** → **Insert as Text** (forces rendering).
* In Outlook Web, you can paste from Word or use dev‑tools “Edit as HTML” as a workaround.

**7) Weekly runbook (Phase 1)**

* **Update Excel** (Sheets 1–4).
* Prompt **Draft newsletter** in the agent.
* Review the HTML draft (tone, counts, highlights).
* **Copy HTML** → paste into Outlook → send.
* **Clear** Excel tables to reset for next week.

**8) Troubleshooting quick fixes**

* **Popup blocked** during connector sign‑in
  + Allow pop‑ups for make.powerautomate.com, refresh, and re‑try.
* **Excel table not found / 404 file**
  + Re‑select **Site / Library / File / Table** in each Excel action.
  + Confirm the ranges are **Tables** and names are exact.
* **Customer Highlights and ROI Highlights look the same**
  + In **Respond to the agent**, ensure customerHighlights maps to **CustomerHighlights → body/value**, and roiHighlightsIndividual maps to **ROIHighlights → body/value**.
* **OneNote HTML conversion error**
  + Use the **Compose** replace‑chain provided; or pass contentHtml directly as meetingNotes.
* **HTML renders as code in Outlook**
  + Use **Insert as Text** method (see step 6).

**9) You’re done — Phase 1 complete**

You now have a working **one‑prompt** newsletter workflow:

* Update Excel → **Draft newsletter** → paste HTML → send.

**Next phases (optional):**

* **Phase 2**: Add a Friday 09:00 reminder flow to nudge updates and drafting.
* **Phase 3**: Add an **employee submission path** (guided Q&A to write to Excel) and an **owner approval + auto‑send** option.

If you want, I can provide:

* A **.zip solution** with the flow pre‑built (parameterized for your SharePoint paths).
* A **styled HTML template** as a starting point (you can tweak colors/logos).