Outlook 'Scan to Email' Add-In – Full Implementation Plan (VSTO + NAPS2 SDK + Optional Nextcloud Server)

Prepared for: Battery Source – CTO (Tyler)
Date: 2025-09-09

Overview

Objective
Build a Windows-only Microsoft Outlook add-in that scans via WIA or TWAIN and auto-attaches the result to the current compose/reply/forward email. Provide (1) rich Profiles with per-profile buttons,
(2) a Review dialog with preview/continue/attach/cancel, (3) temp file handling with auto-purge,
(4) attachment size gates (warn ≥10 MB, block ≥50 MB), (5) telemetry/logging controls, and
(6) an optional central profile management server implemented as a Nextcloud app.

High-Level Decisions
- Platform: VSTO (C#/.NET) add-in for Outlook on Windows.
- Scanning engine: NAPS2 SDK with WIA and TWAIN support; prefer WIA on x64, TWAIN as fallback.
- Ribbon UX: up to 3 “pinned” one-click profile buttons + a dynamic “More Profiles” menu.
- Profiles: JSON-backed, editable in Outlook; Description + ShortDescription for ribbon density.
- Review Dialog: preview pages, rotate/delete/reorder, continue scanning, attach, cancel.
- Temp Storage: per-scan working subfolder under %LOCALAPPDATA% with startup 90-day purge.
- Size Policy: warn ≥10 MB with exact message; block ≥50 MB with exact message.
- Optional Server: Nextcloud app providing server-managed profile templates grouped by “Group” with unique URL per group; clients enroll with a “connect password” to obtain a permanent token.
- Licensing: NAPS2.Sdk = LGPL; NAPS2.Wia = MIT; main NAPS2 app = GPL (not embedded).

Platform Choice

Why VSTO

- Direct access to local hardware APIs and file system; can call NAPS2 SDK directly.
- Simple attachment via Outlook Object Model (MailItem.Attachments.Add(path)).
- Office.js (web add-ins) lack native device access; would require a companion native helper, so VSTO is simpler and more robust for this Windows-only scenario.

Profiles & Ribbon

Profiles & Ribbon

- Each profile has: Description, ShortDescription, provider (Auto/WIA/TWAIN), deviceId (optional), DPI, colorMode, pageSize, paperSource, duplex, output (PDF/JPEG with jpegQuality=75 default, pdfDownsampleDpi, pdfA, fileNameTemplate), hygiene (blankPageDetection with threshold=0.98, deskew, autocrop).
- Pinned Profiles: up to 3 become one-click buttons; labels use ShortDescription when the ribbon is dense; tooltips show the full Description + key settings.
- Dynamic Menu: a splitButton exposes a runtime-populated menu of all profiles (refreshable without restart); includes “Manage Profiles…” entry.
- Server-managed mode: only server profiles can occupy the 3 one-click buttons; user custom profiles appear in the dropdown. Standalone mode: users may pin up to 3 custom profiles.

Scan Pipeline

Scan Pipeline (identical for normal and Test Scan)

1) Resolve effective settings (server template + client overlay OR custom profile).
2) Acquire via NAPS2 (WIA preferred; TWAIN fallback). Support multi-page, ADF/Flatbed, duplex.
3) Save to working folder: single PDF or multiple JPGs (jpegQuality default 75).
4) Review dialog: preview, rotate/delete/reorder, Continue Scanning (append), Attach, Cancel.
5) Pre-attach gate: compute total size; warn ≥10 MB (“Your scanned attachment is over 10 MB. Most recipients can't receive attachments that big. Are you sure you want to continue?”). Block ≥50 MB (“Your scanned attachment is over 50 MB. This is too big to send.”).
6) On Attach: add file(s) to the active compose MailItem; optionally open containing folder.
7) On Cancel: auto-delete the run’s working subfolder.

Temp Storage & Cleanup

Temp Storage, Cleanup & Startup Purge

- Working root: %LOCALAPPDATA%\OutlookScan\Temp\YYYY\MM\<scanGuid>\
- Manifest per run: minimal JSON (profile key, pages, timestamps) to aid diagnostics.
- Startup purge: background task deletes files older than 90 days; prunes empty directories; uses UTC timestamps and ignores per-file errors (best-effort).

Telemetry & Logging

Telemetry & Logging (opt-in)

- Levels: Error / Warn / Info / Debug (default: Info). User-controllable.
- Collected (no PII, no images): mode (server/standalone), profile key, provider used, pages, duration buckets, warning/block triggers, error taxonomy, helper usage (yes/no).
- Diagnostics: “Generate bundle” (logs + run manifests; excludes images).

Helper Process (TWAIN Bridge)

Helper Process (32-bit TWAIN Bridge)

- Purpose: only used when Outlook is x64 and a scanner exposes TWAIN via 32-bit driver only.
- Process: Helper32.exe (32-bit .NET), started on demand; idle timeout to auto-exit.
- Transport: Named Pipes (user-scoped); JSON messages; timeouts and size limits; per-message schema validation.
- Messages:
  - Hello { version } → HelloAck { version }
  - EnumerateDevices { provider:"TWAIN" } → { devices:[{productName, productId}] }
  - StartScan { jobId, settings, device } → Progress {…}* → Complete { files:[paths], pageCount } OR Fail { code }
  - Cancel { jobId } → Canceled { jobId }
- Error codes: FeederEmpty, CoverOpen, UserCancel, DeviceBusy, DriverError, Timeout, UnsupportedCapability.

JSON Schemas – Overview

JSON Schemas (authoritative)

The following schemas define the server manifest (with template settings), the client overlay (device-local bindings), the local custom profile, and the client config (server toggle). These match the behavior you approved: paperSource defaults Auto→ADF→Flatbed (runtime), duplex defaults off, server profiles are read-only on clients; users may duplicate server templates to create custom profiles.

Server Manifest Schema – Summary

SERVER MANIFEST SCHEMA (server-manifest.schema.json)

- Contains: schemaVersion, manifestId, issuedAt, pinnedOrder (≤3), profiles[].
- TemplateSettings cover non-hardware fields (dpi, colorMode, pageSize, paperSource default Auto, duplex default false, output format+quality, hygiene, sizePolicy, defaultSaveLocation).

Server Manifest Schema – JSON

```
{
  "$id": "https://batterysource/schemas/server-manifest.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "OutlookScan Server Manifest",
  "type": "object",
  "required": ["schemaVersion", "manifestId", "issuedAt", "profiles"],
  "properties": {
    "schemaVersion": { "type": "integer", "minimum": 1 },
    "manifestId": { "type": "string", "minLength": 1 },
    "issuedAt": { "type": "string", "format": "date-time" },
    "allowClientCustomProfiles": { "type": "boolean", "default": true },
    "pinnedOrder": {
      "type": "array", "items": { "type": "string" },
      "uniqueItems": true, "maxItems": 3, "default": []
    },
    "profiles": { "type": "array", "minItems": 1, "items": { "refdefs/ServerProfile" } }
  },
  "$defs": {
    "ServerProfile": {
      "type": "object",
      "required": ["key", "name", "template"],
      "properties": {
        "key": { "type": "string", "pattern": "^[a-zA-Z0-9._-]{1,64}$" },
        "name": { "type": "string", "minLength": 1 },
        "description": { "type": "string", "default": "" },
        "shortDescription": { "type": "string", "default": "" },
        "template": { "refdefs/TemplateSettings" }
      }
    },
    "TemplateSettings": {
      "type": "object",
      "properties": {
        "dpi": { "type": "integer", "minimum": 75, "maximum": 1200, "default": 300 },
        "colorMode": { "type": "string", "enum": ["Color","Grayscale","BlackWhite"], "default":
"Grayscale" },
        "pageSize": {
          "oneOf": [
            { "type": "string", "enum": ["Letter","Legal","A4","A5","Auto"] },
            { "type": "object", "required": ["unit","width","height"],
              "properties": {
                "unit": { "type": "string", "enum": ["inch","mm"] },
                "width": { "type": "number", "exclusiveMinimum": 0 },
                "height": { "type": "number", "exclusiveMinimum": 0 } },
              "additionalProperties": false }
          ],
          "default": "Letter"
        },
        "paperSource": {
          "type": "string", "enum": ["Auto","ADF","Flatbed"], "default": "Auto",
          "description": "Runtime fallback: Auto → ADF if available → Flatbed."
        },
        "duplex": { "type": "boolean", "default": false },
        "output": {
          "type": "object", "required": ["format"],
          "properties": {
            "format": { "type": "string", "enum": ["PDF","JPEG"] },
            "jpegQuality": { "type": "integer", "minimum": 1, "maximum": 100, "default": 75 },
            "pdfDownsampleDpi": { "type": "integer", "minimum": 72 },
            "pdfA": { "type": "boolean", "default": false },
            "fileNameTemplate": { "type": "string", "default": "Scan_{yyyyMMdd_HHmmss}" }
          },
          "additionalProperties": false
        },
        "hygiene": {
          "type": "object",
          "properties": {
            "blankPageDetection": {
              "oneOf": [
                { "type": "boolean" },
                { "type": "object", "required": ["enabled"],
                  "properties": { "enabled": { "type": "boolean" }, "threshold": { "type": "number", "minimum": 0, "maximum": 1, "default": 0.98 } },
                  "additionalProperties": false }
              ],
              "default": false
            },
            "deskew": { "type": "boolean", "default": false },
            "autocrop": { "type": "boolean", "default": false }
          },
          "additionalProperties": false
        },
        "sizePolicy": {
          "type": "object",
          "properties": {
            "warnTotalBytes": { "type": "integer", "default": 10485760 },
            "blockTotalBytes": { "type": "integer", "default": 52428800 }
          },
          "additionalProperties": false,
          "default": { "warnTotalBytes": 10485760, "blockTotalBytes": 52428800 }
        },
        "defaultSaveLocation": { "type": "string" }
      },
      "additionalProperties": false
    }
  },
  "additionalProperties": false
```

```
}
```

Client Overlay Schema – JSON

CLIENT OVERLAY SCHEMA (client-overlay.schema.json)

- Binds a server template to local device choices per machine.

```
{
  "$id": "https://batterysource/schemas/client-overlay.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "OutlookScan Client Overlay",
  "type": "object",
  "required": ["schemaVersion","machineId","overrides"],
  "properties": {
    "schemaVersion": { "type": "integer", "minimum": 1 },
    "machineId": { "type": "string", "minLength": 1 },
    "overrides": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["templateKey"],
        "properties": {
          "templateKey": { "type": "string" },
          "providerPreferred": { "type": "string", "enum": ["Auto","WIA","TWAIN"], "default": "Auto" },
          "deviceId": { "type": "string" },
          "paperSourcePreferred": { "type": "string", "enum": ["Auto","ADF","Flatbed"], "default": "Auto" },
          "duplexPreferred": { "type": "boolean", "default": false }
        },
        "additionalProperties": false
      },
      "default": []
    }
  },
  "additionalProperties": false
}
```

Custom Profile Schema – JSON

CUSTOM PROFILE SCHEMA (custom-profile.schema.json)
- Local profiles (standalone mode) or duplicates of server templates.

```
{
  "$id": "https://batterysource/schemas/custom-profile.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "OutlookScan Custom Profile",
  "type": "object",
  "required": ["schemaVersion","key","name","settings"],
  "properties": {
    "schemaVersion": { "type": "integer", "minimum": 1 },
    "key": { "type": "string", "pattern": "^[a-zA-Z0-9._-]{1,64}$" },
    "name": { "type": "string", "minLength": 1 },
    "description": { "type": "string", "default": "" },
    "shortDescription": { "type": "string", "default": "" },
    "originTemplateKey": { "type": "string" },
    "pinned": { "type": "boolean", "default": false },
    "pinnedOrder": { "type": "integer", "minimum": 1, "maximum": 3 },
    "settings": {
      "allOf": [
        { "$ref": "https://batterysource/schemas/server-manifest.schema.json#/$defs/TemplateSettings" },
        { "type": "object", "properties": {
            "providerPreferred": { "type": "string", "enum": ["Auto","WIA","TWAIN"], "default": "Auto" },
            "deviceId": { "type": "string" } },
          "additionalProperties": false }
      ]
    }
  },
  "additionalProperties": false
}
```

Client Config Schema – JSON

CLIENT CONFIG SCHEMA (client-config.schema.json)

- Toggles server mode and points to base URL; used by admin deployment.

```
{
  "$id": "https://batterysource/schemas/client-config.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "OutlookScan Client Config",
  "type": "object",
  "required": ["enabled"],
  "properties": {
    "enabled": { "type": "boolean", "default": false },
    "baseUrl": { "type": "string", "format": "uri" },
    "tenantId": { "type": "string" },
    "verifySignature": { "type": "boolean", "default": true },
    "updateIntervalHours": { "type": "integer", "minimum": 1, "default": 12 }
  },
  "additionalProperties": false
}
```

Nextcloud Server – Architecture

Nextcloud Server App (outlookscan_profiles)

- Entities: Groups (slug+name), Profiles, Group→Profiles, Group pin order (0..3), Group connect secret, Clients (machineName, token hash/prefix, lastSeen, revoked).
- Client Enrollment: POST /api/v1/groups/{slug}/connect { machineName, connectPassword } → returns permanent clientToken + manifestUrl. Rotating connectPassword does not affect existing clients.
- Manifest Fetch: GET /api/v1/groups/{slug}/manifest with X-Client-Token; returns signed JSON + ETag.
- Admin OCS endpoints (auth required): CRUD groups/profiles; set pin order; list/revoke clients; set connect password; manage signing key; choose which NC group has GUI access.
- Admin GUI Access: restricted to members of a custom NC group (e.g., “OutlookScan Admins”).

Endpoints

Endpoint Blueprint
Client (no login):
- POST /apps/outlookscan_profiles/api/v1/groups/{slug}/connect
- GET /apps/outlookscan_profiles/api/v1/groups/{slug}/manifest

Admin (OCS, NC auth required + OCS-APIRequest: true):
- /ocs/v2.php/apps/outlookscan_profiles/profiles (GET/POST/PUT/DELETE)
- /ocs/v2.php/apps/outlookscan_profiles/groups (GET/POST/PUT/DELETE)
- /ocs/v2.php/apps/outlookscan_profiles/pinorder/{group} (GET/PUT)
- /ocs/v2.php/apps/outlookscan_profiles/clients (GET) & /clients/{id} (DELETE revoke)

Database Schema

Database Tables (prefix oc_oscan_...)

- groups(id, slug UNIQUE, name, created_at)
- profiles(key PK, name, description, short_description, template_json, updated_at, updated_by)
- group_profiles(group_id FK, profile_key FK, UNIQUE(group_id, profile_key))
- group_pinorder(group_id FK, pos INT, profile_key FK, UNIQUE(group_id,pos))
- group_secrets(group_id FK, connect_secret_hash, rotated_at)
- clients(id PK, group_id FK, machine_name, token_hash, token_prefix, created_at, last_seen_at, revoked BOOL)

Security

Security & Trust

- HTTPS only; tokens are bearer credentials; store only hashes (Argon2id) + short prefixes for display.
- Rate-limit /connect; lockout on repeated failures; audit log admin actions.
- Sign manifests (JWS); client verifies signature before applying.
- ETag/If-None-Match for efficient startup checks.

Acceptance Criteria & Handoff

Acceptance Criteria (abridged)
1) VSTO add-in: 3 pinned buttons + dynamic menu; full Scan→Review→Attach; Test Scan uses full pipeline.
2) Temp handling: per-scan folder; cancel deletes; startup purge >90 days.
3) Size gates: warn ≥10 MB (exact message); block ≥50 MB (exact message).
4) Telemetry/logging with level control; diagnostics bundle.
5) Server mode: groups with unique URLs; connect password → permanent token; rotate password without dropping existing clients; admin can list/revoke clients.
6) Admin GUI restricted to a custom NC group; OCS endpoints for automation.

What to paste into a new ChatGPT "build" chat

- State the goal: "Build an Outlook VSTO add-in using NAPS2 SDK with WIA/TWAIN scanning, profiles as defined below, and an optional Nextcloud server for central profile management."
- Provide this PDF (or paste the JSON Schemas and endpoint list).
- Ask for: (a) VSTO project scaffold with Ribbon (3 pinned + dynamic menu), (b) ProfileManager (schemas, merge logic, server cache & overlay), (c) Startup services (server sync + 90-day purge), (d) Scan pipeline wrapper around NAPS2 with Review dialog, (e) Attachment size-gate hook, (f) Helper32 IPC skeleton, (g) Nextcloud app skeleton (routes/controllers/migrations + Vue admin), (h) unit tests for merge & purge.

Licensing & Next Steps

Licensing Notes

- NAPS2.Sdk: LGPL – dynamically link; include license notice; provide upstream source/binaries used.
- NAPS2.Wia: MIT – attribution only.
- Main NAPS2 desktop app: GPL – not embedded in the add-in.

Recommended Build Order

1) Schemas & POCOs → ProfileManager (merge/validation, pinning policy).
2) Ribbon (pinned buttons + dynamic menu; density-aware labels; tooltips).
3) Startup services: server sync (ETag + JWS verify) and 90-day purge.
4) Scan pipeline (NAPS2) + Review dialog + size gates + cancel cleanup.
5) Helper32 + IPC (conditional on TWAIN 32-bit need).
6) Nextcloud server app (DB migrations, controllers, OCS endpoints, Vue admin).
7) Settings UI (telemetry/logging level, export/import profiles, server wizard).

Risks & Mitigations

- TWAIN 32-bit on x64 Outlook → Helper32 bridge.
- Large attachments → size gates + guidance to lower DPI / JPEG per page.
- Device not found → device picker + overlay update; clear error messages.
- Server outage → cached manifest + offline banner; retry backoff.