

Email System Runbook

Email System Runbook

Overview

Centralized email system for ALAI/BasicAS. All outbound email goes through IMAP/SMTP (one.com + domeshop), with a single audit database tracking everything.

Accounts

Account	Email	Provider	Usage
john	john@alai.no	Migadu	Primary business email
info	info@alai.no	Migadu	General inquiries
alai	john@alai.no	Migadu	ALAI branded, signing emails

Credentials stored in **Vaultwarden** (`bw get item "Migadu - john@alai.no"`).

How to Send Email

Option 1: MCP (from Claude session — PREFERRED)

```
mcp_email_email_send({
  account_name: "john",
  to: "client@example.com",
  subject: "Subject",
  body: "Body text",
  body_type: "html",           // or "plain"
```

```
attachments: [{path: "/absolute/path/file.pdf"}] // optional
})
```

Option 2: CLI (from scripts, daemons, agents)

```
node ~/system/tools/mail-native.js send \  
  --to client@example.com \  
  --subject "Subject" \  
  --body "Body text" \  
  --account john \  
  --attachment /path/to/file.pdf # optional, comma-separated for multiple
```

Option 3: Signing Emails (DocuSeal)

```
node ~/system/tools/send-signing-email.js send <template_id> '<signer_json>' --test
```

How to Read Email

MCP (preferred)

```
mcp_email_emails_find({account_name: "john", query: "invoice", limit: 10})  
mcp_email_email_respond({email_id: "12345", body: "Reply text"})
```

CLI

```
node ~/system/tools/mail-native.js search "invoice" --account john --limit 20  
node ~/system/tools/mail-native.js read <uid> --account john  
node ~/system/tools/mail-native.js unread --account john  
node ~/system/tools/mail-native.js reply <uid> --body "Reply text"  
node ~/system/tools/mail-native.js forward <uid> --to other@email.com  
node ~/system/tools/mail-native.js attachment <uid> --save /tmp/downloads
```

Email Audit (Single Source of Truth)

Database: ~/system/databases/email-audit.db

Every outbound email is logged here, regardless of send path:

- **mail-native.js** — hard require, logs on every send/reply/forward
- **MCP bridge** — logs via JS module + Python hook (dedup by message_id)
- **signing email** — logs via JS module
- **Hook safety net** — `email-outbox-logger.py` catches MCP sends even if JS fails

Quick Commands

```
node ~/system/tools/email-audit.js recent           # Last 10 sent emails
node ~/system/tools/email-audit.js find "client name" # Search all emails
node ~/system/tools/email-audit.js find "invoice" --days 30 # Last 30 days
node ~/system/tools/email-audit.js stats --days 7      # Stats by tool/account
node ~/system/tools/email-audit.js health             # System health check
node ~/system/tools/mail-native.js audit --days 30    # Audit from CLI
node ~/system/tools/mail-native.js sent --account john # IMAP Sent folder
```

Architecture

Send paths:

```
MCP (email_send/respond) ─┐
mail-native.js CLI ───┐───┤───> email-audit.db (single source of truth)
send-signing-email.js ───┘
Hook (email-outbox-logger) ─┘ (safety net, dedup by message_id)
```

DEPRECATED Tools (DO NOT USE)

Tool	Replacement
email.js	mail-native.js
email-monitor.js	MCP bridge
email-outbox.db	email-audit.db
Inline SMTP scripts	BLOCKED by bash-security-gate.py

Email-to-Task Integration

Automatic Task Creation

The email-agent daemon (`~/system/daemons/email-agent.js`) automatically creates MC tasks for ACTION emails:

- Classifies incoming emails as ACTION/FYI/SPAM
- Creates MC task for ACTION emails via `email-to-task.js`
- Links email record to MC task via `mc_task_id` column

Backlog Processing (MC #9269)

On daemon startup, email-agent processes ACTION emails that missed MC task creation:

```
SELECT id, message_id, account, from_addr, from_name, subject, date, action_needed, summary,
priority
FROM emails
WHERE classification = 'ACTION' AND mc_task_id IS NULL
ORDER BY date DESC
```

How it works:

1. Query runs once per daemon startup (before new email processing)
2. For each backlog email: calls `email-to-task.js` with full context
3. Extracts MC task ID from output (`MC task #1234`)
4. Updates email record: `UPDATE emails SET mc_task_id = ? WHERE id = ?`
5. Non-blocking: errors logged but don't crash daemon

Idempotency: Duplicate detection handled by `mc.js` (same title + <24h → link to existing task)

Removed 24h cutoff (2026-04-25):

- **Old logic:** Only process emails from last 24h
- **New logic:** Process ALL ACTION emails without `mc_task_id` (no time filter)
- **Reason:** 78 ACTION emails backlogged over weeks — cutoff prevented recovery

Manual Backfill Procedure

If ACTION emails accumulate without MC tasks:

```
# 1. Check backlog count
sqlite3 ~/system/databases/email-inbox.db \  
"SELECT COUNT(*) FROM emails WHERE classification='ACTION' AND mc_task_id IS NULL;"
```

```
# 2. Restart daemon (triggers backlog processing)
launchctl kickstart -k gui/$(id -u)/com.john.email-agent

# 3. Verify log
tail -50 ~/system/logs/email-agent.log | grep -A10 "BACKLOG PROCESSING"

# 4. Check result
sqlite3 ~/system/databases/email-inbox.db \
  "SELECT COUNT(*) FROM emails WHERE classification='ACTION' AND mc_task_id IS NULL;"
```

Reference: `/tmp/mc-9269-completion-report.md` — 2026-04-25 backfill (78 emails → 69 MC tasks)

Email Tracker (Open/Click Tracking)

Service: `com.john.email-tracker` (LaunchAgent, KeepAlive) **File:** `~/system/tools/email-tracker.js`
Port: 3456 (127.0.0.1 only) **Logs:** `~/system/logs/email-tracker-stdout.log` / `email-tracker-stderr.log`

Modes

```
node ~/system/tools/email-tracker.js          # server mode (default, daemon)
node ~/system/tools/email-tracker.js stats    # print DB counts, exit 0
node ~/system/tools/email-tracker.js tail     # stream new emails as log
```

Endpoints

Endpoint	Description
<code>GET /health</code>	<code>{"ok":true}</code> liveness check
<code>GET /api/dashboard</code>	JSON stats: by_status, by_class, tracking events
<code>GET /track/:emailId/open</code>	Log open event (returns 1x1 GIF)
<code>GET /track/:emailId/click</code>	Log click event

DB Tables

- `emails` — existing table, queried for stats
- `email_tracking_events` — created on first run (`id, email_id, event_type, ip, user_agent, created_at`)

Commands

```
# Quick stats
node ~/system/tools/email-tracker.js stats

# Live dashboard
curl -s http://127.0.0.1:3456/api/dashboard | jq .

# Reload daemon (after file changes or crash)
launchctl kickstart -k gui/$(id -u)/com.john.email-tracker

# Check status
launchctl print gui/$(id -u)/com.john.email-tracker | grep -E "state|pid"
```

Troubleshooting

Email not in audit

1. Check `node email-audit.js recent` — is it really missing?
2. Check MCP bridge log: `tail ~/system/logs/email-mcp-bridge.log`
3. Check mail-native log: `tail ~/system/logs/mail-native.log`
4. Run `node email-audit.js health` — any warnings?

SMTP connection fails

1. Check vault: `bw get item "Migadu - john@alai.no" --session $(cat /tmp/bw-session) | jq .login.username`
2. Test: `node mail-native.js test --account john`
3. one.com rate limits: wait 5 min, retry

Attachments not working

1. Verify file exists: `ls -la /path/to/file`
2. Use absolute paths only
3. Max attachment size: ~25MB (one.com limit)
4. CLI: `--attachment /path/file1.pdf,/path/file2.pdf` (comma-separated)
5. MCP: `attachments: [{path: "/abs/path"}]` (array of objects)

ACTION email has no MC task

Symptoms: Email classified as ACTION but `mc_task_id IS NULL` in database

Diagnosis:

```
# Check specific email
sqlite3 ~/system/databases/email-inbox.db \
  "SELECT id, from_addr, subject, classification, mc_task_id FROM emails WHERE id =
  <email_id>;"

# Check backlog count
sqlite3 ~/system/databases/email-inbox.db \
  "SELECT COUNT(*) FROM emails WHERE classification='ACTION' AND mc_task_id IS NULL;"
```

Fix:

1. Check daemon is running: `launchctl list | grep email-agent`
2. Check daemon log: `tail -100 ~/system/logs/email-agent.log`
3. Trigger backlog processing: `launchctl kickstart -k gui/$(id -u)/com.john.email-agent`
4. If still NULL after daemon cycle:
 - Manual task creation: `node ~/system/tools/email-to-task.js --from "." --subject "." --message-id "."`
 - Update email record: `sqlite3 ~/system/databases/email-inbox.db "UPDATE emails SET mc_task_id = <task_id> WHERE id = <email_id>;"`

Revision #8

Created 2026-02-20 09:38:31 UTC by John

Updated 2026-06-21 20:01:51 UTC by John