

# Email Agent Runbook

# Email Agent Runbook

**Service:** Email Agent Daemon

**Location:** `~/system/daemons/email-agent.js`

**LaunchAgent:** `com.john.email-agent`

**Interval:** Every 5 minutes (300s)

**Last Updated:** 2026-04-15

---

## 1. Architecture

### What It Does

The Email Agent is a 24/7 daemon that:

- Fetches unseen emails from **6 IMAP accounts** every 5 minutes
- Classifies emails using VIP bypass → quick filter → Ollama (llama3.1:8b, \$0 cost)
- Creates Mission Control tasks for ACTION-worthy emails
- Auto-archives INFO and SPAM emails
- Downloads attachments for CEO-forwarded emails
- Logs all activity to HiveMind and JSONL results

### Accounts Monitored

Account Key	Email Address	Bitwarden Vault Name
<code>john</code>	john@basicconsulting.no	Email - john@basicconsulting.no
<code>info</code>	info@basicconsulting.no	Email - info@basicconsulting.no
<code>alai</code>	john@alai.no	Email - john@alai.no
<code>alem</code>	alem@alai.no	Email - alem@alai.no
<code>dev</code>	dev@alai.no	Email - dev@alai.no
<code>gmail</code>	alembasic@gmail.com	Email - alembasic@gmail.com

# Classification Pipeline

1. **VIP Bypass:** Emails from CEO/family → forced to `ACTION/high`, label: `CEO FORWARD`
2. **Quick Filter:** Pattern-based detection for OWN emails and known SPAM
3. **Ollama Classification:** Remaining emails sent to local llama3.1:8b model
4. **Circuit Breaker:** Falls back to pattern heuristics if Ollama is down (3 failure threshold)

## VIP Senders (CEO Bypass List)

Emails from these addresses **bypass all filters** and are always classified as `ACTION/high` with label `CEO FORWARD`:

- `alem@alai.no`
- `alem@basicconsulting.no`
- `alem.basic@gmail.com`
- `alembasic@gmail.com`
- `sibilabasic@gmail.com` (CEO's wife)
- `riadbasic007@gmail.com` (CEO's brother)

## Transport: Himalaya Adapter

The daemon uses `~/system/tools/himalaya-adapter.js`, which wraps the Rust-based `himalaya` CLI (`/opt/homebrew/bin/himalaya`).

**Config:** `~/config/himalaya/config.toml` — all 6 accounts configured.

---

## 2. Credentials

### Bitwarden Storage

All email accounts are stored in Bitwarden with vault item names following the pattern: `Email - <address>`.

### Gmail Account (Special Configuration)

The Gmail account (`alembasic@gmail.com`) uses **App Password authentication** (not the regular Google account password).

**Bitwarden Item:** `Email - alembasic@gmail.com`

**Custom Fields in Vault:**

- `imap_host` = `imap.gmail.com`
- `imap_port` = `993`
- `password` = **App Password** (16-character token from Google)

## Himalaya Config

File: `~/.config/himalaya/config.toml`

Contains 6 account blocks with IMAP/SMTP settings. Credentials are loaded from Bitwarden at runtime via `mail-native.js`.

---

## 3. How to Verify

### Is the Daemon Running?

```
launchctl list | grep email-agent
# Expected output: PID + exit status 0
# Example: 12345 0 com.john.email-agent
```

### Last Heartbeat (Should Be < 10 Minutes Ago)

```
cat ~/system/logs/email-agent-heartbeat.txt
# Shows timestamp of last successful run
```

### Recent Activity Log

```
tail -20 ~/system/logs/email-agent-launchd.log
# Should show recent classification activity like:
# {"timestamp":"2026-04-15T13:49:06.450Z","service":"email-agent","level":"info","message":"Classifying via Ollama: ..."}
```

### Pending Emails (Email Inbox Tool)

```
node ~/system/tools/email-inbox.js pending
# Lists emails waiting for classification or action
```

# Daemon Status (Full Details)

```
launchctl print gui/$(id -u)/com.john.email-agent  
# Shows full launchd status, last run time, exit codes
```

## 4. Troubleshooting

### Problem: Daemon Dead (MODULE\_NOT\_FOUND Error)

#### Symptom:

```
tail -20 ~/system/logs/email-agent-launchd-error.log  
# Shows: Error: Cannot find module '~/system/tools/himalaya-adapter'
```

**Root Cause:** The `himalaya-adapter.js` file was accidentally archived or deleted.

#### Fix:

1. Verify the file exists: `ls -lh ~/system/tools/himalaya-adapter.js`
2. If missing, restore from `~/system/tools/archive/` or Git history
3. Restart the daemon:

```
launchctl unload ~/Library/LaunchAgents/com.john.email-agent.plist  
launchctl load ~/Library/LaunchAgents/com.john.email-agent.plist
```

4. Verify restart: `launchctl list | grep email-agent`

### Problem: Gmail "Unknown Account" Error

#### Symptom:

```
Error: Unknown account: gmail. Available: john, info, alai, alem, dev
```

**Root Cause:** The `gmail` key is missing from the `VAULT_NAMES` object in `~/system/tools/mail-native.js`.

#### Fix:

1. Open `~/system/tools/mail-native.js`
2. Locate the `VAULT_NAMES` object (around line 20)
3. Add the gmail entry:

```
const VAULT_NAMES = {
  john: 'Email - john@basicconsulting.no',
  info: 'Email - info@basicconsulting.no',
  alai: 'Email - john@alai.no',
  alem: 'Email - alem@alai.no',
  dev: 'Email - dev@alai.no',
  gmail: 'Email - alembasic@gmail.com' // Add this line
};
```

4. Save and reload daemon

---

## Problem: Gmail Hanging Daemon (High CPU/Memory)

### Symptom:

- Multiple overlapping `email-agent` processes running
- 400%+ CPU usage (seen in `top`)
- Email agent not completing runs

**Root Cause:** Gmail IMAP fetch is hanging indefinitely, causing overlapping daemon instances.

### Fix:

1. Identify stuck process:

```
ps aux | grep email-agent
```

2. Kill the stuck process gracefully:

```
kill -QUIT <PID>
# Or if unresponsive:
kill -9 <PID>
```

3. Unload and reload daemon:

```
launchctl unload ~/Library/LaunchAgents/com.john.email-agent.plist
launchctl load ~/Library/LaunchAgents/com.john.email-agent.plist
```

---

# Problem: Vault Credentials Unavailable (Circuit Breaker Triggered)

## Symptom:

```
Error: Bitwarden session not available
# 0r: Circuit breaker OPEN for account: john
```

**Root Cause:** Bitwarden CLI session expired or `/tmp/bw-session` is empty.

## Fix:

1. Check session file:

```
cat /tmp/bw-session
# Should contain a session token string
```

2. If empty, unlock Bitwarden and regenerate session:

```
bw unlock --raw > /tmp/bw-session
# Enter master password when prompted
```

3. Verify session works:

```
bw get item "Email - john@basicconsulting.no" --session $(cat /tmp/bw-session)
```

4. Circuit breaker will reset automatically on next successful run (backoff resets after threshold period)

---

# Problem: Alem's Emails Not Showing as ACTION

**Symptom:** Emails from CEO are classified as INFO or SPAM instead of ACTION/high.

**Root Cause:** VIP\_SENDERS list is incomplete or outdated.

## Fix:

1. Open `~/system/daemons/email-agent.js`
2. Locate the `VIP_SENDERS` array (around line 92)
3. Ensure all Alem's addresses are present:

```
const VIP_SENDERS = [  
  'alem@alai.no',  
  'alem@basicconsulting.no',  
  'alem.basic@gmail.com',  
  'alembasic@gmail.com',  
  'sibilabasic@gmail.com',  
  'riadbasic007@gmail.com'  
];
```

4. Save and reload daemon

---

## Problem: Ollama Circuit Breaker Open (Fallback Mode)

### Symptom:

```
WARN: Ollama circuit breaker OPEN – using pattern heuristic
```

**Root Cause:** Ollama service is down or unresponsive (3+ consecutive failures).

### Fix:

1. Check Ollama service:

```
curl http://localhost:11434/api/tags  
# Should return JSON list of models
```

2. If unresponsive, restart Ollama:

```
brew services restart ollama  
# Or manually:  
ollama serve
```

3. Circuit breaker will auto-reset after backoff period (starts at 10s, max 5 minutes)
  4. Emails will still be processed using pattern-based heuristics during circuit breaker OPEN state
- 

## 5. Gmail App Password Setup

If the Gmail App Password needs to be regenerated (e.g., after credential rotation or security incident):

1. Go to <https://myaccount.google.com/apppasswords> (must be logged in as alembasic@gmail.com)
2. Click **Generate**
3. Select app: **Mail**
4. Select device: **Mac** (or custom name like "IMAP Daemon")
5. Copy the 16-character App Password (no spaces)
6. Update Bitwarden:

```
bw get item "Email - alembasic@gmail.com" --session $(cat /tmp/bw-session) | \
jq '.login.password = "<NEW_APP_PASSWORD>"' | \
bw encode | \
bw edit item $(bw get item "Email - alembasic@gmail.com" --session $(cat /tmp/bw-
session) | jq -r .id) --session $(cat /tmp/bw-session)
```

Or update manually via Bitwarden web vault.

7. Reload daemon:

```
launchctl unload ~/Library/LaunchAgents/com.john.email-agent.plist
launchctl load ~/Library/LaunchAgents/com.john.email-agent.plist
```

## 6. Key Files and Locations

File	Purpose
<code>~/system/daemons/email-agent.js</code>	Main daemon script
<code>~/system/tools/mail-native.js</code>	VAULT_NAMES map + credential loader
<code>~/system/tools/himalaya-adapter.js</code>	Himalaya CLI wrapper (IMAP/SMTP)
<code>~/config/himalaya/config.toml</code>	Himalaya account configuration
<code>~/Library/LaunchAgents/com.john.email-agent.plist</code>	LaunchAgent config (5-minute interval)
<code>~/system/logs/email-agent-launchd.log</code>	Daemon stdout log
<code>~/system/logs/email-agent-launchd-error.log</code>	Daemon stderr log
<code>~/system/logs/email-agent-heartbeat.txt</code>	Last successful run timestamp
<code>~/system/logs/email-triage-results.jsonl</code>	JSONL log of all classifications
<code>/tmp/bw-session</code>	Bitwarden CLI session token

# 7. Escalation

If the daemon is down for > 30 minutes and troubleshooting steps do not resolve:

1. Check `email-agent-launchd-error.log` for stack traces
2. Capture full logs:

```
tail -100 ~/system/logs/email-agent-launchd.log > /tmp/email-agent-debug.log
tail -100 ~/system/logs/email-agent-launchd-error.log >> /tmp/email-agent-debug.log
launchctl print gui/$(id -u)/com.john.email-agent >> /tmp/email-agent-debug.log
```

3. Slack alert to `#ops`:

```
node ~/system/tools/slack.js send ops "@john Email Agent daemon DOWN for 30+ minutes.
Logs: /tmp/email-agent-debug.log"
```

4. Fallback: manually check inboxes via webmail until daemon is restored

---

**Document Status:**  Production

**Owner:** John (primary agent)

**Last Incident:** 2026-02-25 — MODULE\_NOT\_FOUND (himalaya-adapter archived)

**Last Review:** 2026-04-15

---

Revision #2

Created 2026-04-15 13:56:23 UTC by John

Updated 2026-05-31 20:06:03 UTC by John