

Email Meeting Scheduler (EMS)

Detailed Functional & Validation Guide • **Version 1.3** • Now smarter, faster, and ready for global teams.

WHAT'S NEW • v1.3

EMS just leveled up. **v1.3** introduces three powerful upgrades that turn your inbox into a true scheduling co-pilot: it now **reads time zones straight from the email**, it **invites every attendee mentioned in the message**, and it **only suggests slots inside your configured working hours**. No more 6 AM proposals. No more manual CC chains. No more time-zone math.

Capability	Before (v1.2)	Now (v1.3)
Time zone handling	Used a single Default Time Zone EV for all replies.	Reads UTC offset / IANA TZ directly from the email body on every request.
Attendees	Booked the meeting with the sender only.	Auto-detects multiple attendees in To/Cc/body and adds them to the invite.
Alternative slots	Generated within a fixed Workday window.	Strictly respects per-requester working hours from the EV — no out-of-hours suggestions.

1. Purpose & Scope

EMS is a Power Automate cloud flow that converts free-form email requests into Outlook calendar outcomes. It honors the exact time typed by the requester, checks availability across all attendees, and either books the event or replies with smart alternative slots. Display times are now intelligently rendered using the time zone **detected in the email itself**, falling back to your configured default only when nothing is found.

This guide explains how EMS parses text, constructs the event window, determines conflicts across multiple calendars, selects alternatives within working hours, and renders outbound emails.

2. High-Level Architecture

- **Trigger:** Outlook “When a new email arrives (V2).”
- **AI Builder:** Extracts proposed date/time, duration, time zone, attendees, and a normalized local ISO without conversions.
- **Dataverse:** Stores Environment Variables (EVs); AI Builder uses Dataverse for the connection.
- **Outlook:** Calendar View query for conflicts (per attendee), event creation with full attendee list, and reply emails.
- **NEW → Dynamic TZ Resolver:** A new step compares the email's detected time zone against the Default TZ EV and uses whichever the requester actually intended.

- **NEW → Attendee Resolver:** Combines To/Cc recipients with any email addresses or @mentions extracted from the body, deduplicates them, and feeds them into the event creation step.

3. Prerequisites & Package

Requirement	Details
Environment	Power Platform environment with Dataverse.
Role	Reviewer/admin account with Exchange Online mailbox.
AI Builder Credits	Required for the extraction call. Ensure your tenant has credits.
Connections	Office 365 Outlook (calendar + mail), Microsoft Dataverse.

Solution **package** **URL:**
https://emailautomation001.blob.core.windows.net/emailmeetingmanager/EmailMeeting_AppSource.zip

4. Install the Solution

- Download the ZIP from the link above.
- Power Apps → Solutions → Import → upload the ZIP and proceed.
- After import, verify connections show **Connected** (Outlook, Dataverse).
- Open the **EMS - Schedule Meeting** flow and turn it on.

5. Environment Variables (EV) — Detailed Reference

EVs let you tune EMS without editing the flow. v1.3 keeps every existing EV fully backwards-compatible — your old configuration just works — and introduces smarter behavior on top.

5.1 Default Year (`suhaib_Default_Year`)

Purpose: Year applied when the request omits a year. EMS overwrites the year of the parsed date with this EV. If *Roll Past to Next Year* is true and the resulting date is in the past, EMS increments by one year. **Sample:** 2026. **Example:** '10 Oct 2 pm' → 2026-10-10 14:00 local.

5.2 Default Meeting Duration (mins) (`suhaib_DefaultMeetingDuration`)

Purpose: Fallback duration (integer minutes) when the email doesn't specify one. **Sample:** 35. **Example:** 'meet at 14:00' → 35 minutes.

5.3 Default Time Zone (`suhaib_DefaultTimeZone`)

Purpose: Fallback time zone for displaying meeting times in EMS reply emails. It does *not* change the actual calendar invite — Outlook auto-converts the invite to each attendee's local zone. **Sample:** Pakistan Standard Time.

NEW IN v1.3 • Dynamic TZ takes precedence

If the email contains a usable time zone (PST, CET, UTC+5, +05:00, Asia/Karachi, Pakistan Standard Time, etc.), v1.3 uses **that** for the reply rendering. The EV is now a graceful fallback, not the master switch. See §6 for the full resolution order.

5.4 Workday Start / End Local (`suhaib_WorkdayStartLocal`, `suhaib_WorkdayEndLocal`)

Purpose: Earliest and latest local wall-clock times (HH:mm) used to generate alternative slots. **Samples:** 09:00 and 18:00.

NEW IN v1.3 • Working hours are now strictly enforced

v1.3 will **never** propose a slot that starts before Workday Start or ends after Workday End — even if there's free time outside those bounds. If the requested day has no valid window, EMS rolls to the next working day automatically.

5.5 Suggestion Count (`suhaib_SuggestionsCount`)

Purpose: Number of alternative slots in busy replies. **Sample:** 3.

5.6 Suggestion Lead Time (mins) (`suhaib_SuggestionLeadTimeMins`)

Purpose: Minimum minutes between 'now' and the first suggested start, so attendees aren't ambushed. **Sample:** 60.

5.7 Roll Past to Next Year (`suhaib_RollPastToNextYear`)

Purpose: Boolean. If the computed date is in the past, roll it forward one year before booking or suggesting. **Sample:** true.

5.8 Extract Schema JSON (`suhaib_ExtractSchemaJSON`)

Purpose: JSON schema that AI Builder uses to extract fields. **v1.3 adds two fields:** *attendees* and *detected_timezone* (see §7). Replace your existing schema with the v1.3 version below.

5.9 Booked Reply HTML / Busy Reply HTML (suhaib_BookedReplyHTML, suhaib_BusyReplyHTML)

Purpose: HTML templates for booked and busy outcomes. **New tokens in v1.3:** {AttendeesList}, {ResolvedTZ}, {WorkdayStart}, {WorkdayEnd}.

6. NEW — Dynamic Time Zone Reading From the Email

Until v1.2, every reply was rendered in the Default Time Zone EV regardless of where the requester actually was. That worked for single-region teams, but it showed its age the moment a colleague in Berlin wrote ‘10 Nov 14:00 CET’ and got a reply rendered in Pakistan time.

v1.3 fixes this with a dedicated **Dynamic TZ Resolver** step that runs right after AI Builder extraction.

6.1 What EMS now recognizes

- **Abbreviations:** PKT, PST, EST, CET, GMT, BST, JST, IST ...
- **IANA names:** Asia/Karachi, Europe/Berlin, America/New_York ...
- **Windows display names:** Pakistan Standard Time, India Standard Time ...
- **Numeric offsets:** +05:00, GMT+5, UTC-7, -0800.

6.2 Resolution order

1	Time zone explicitly written in the email body.
2	Sender's mailbox time zone (when readable from the Outlook profile).
3	Default Time Zone EV — final safety net.

6.3 Important: time zone affects display, not the invite

The calendar event itself is always stored in UTC. The detected time zone changes only how the meeting time is *rendered* in the EMS reply email. Outlook still auto-converts the invite to each attendee's local zone on receipt — exactly as before.

Example

A colleague in Toronto sends: “Can we sync on 11 Nov 10:00 EST?”
EMS detects **EST**, books the event at the correct UTC instant, and the reply email confirms **10:00 EST (15:00 UTC)**. The Toronto attendee sees 10:00 EST; the Karachi attendee sees 20:00 PKT in their Outlook — automatically.

7. NEW — Multiple Attendees, Auto-Detected

EMS used to book the meeting with just the sender. v1.3 builds a complete attendee list by combining three sources, deduplicating addresses, and adding everyone to the Outlook event as required attendees.

7.1 Where attendees come from

- **To & Cc fields** of the original email (excluding your own mailbox).
- **Email addresses** embedded in the body (regex extraction).
- **Named people** resolved via AI Builder when the email says things like ‘loop in Sara and Bilal’ or ‘with the design team’. Names are resolved against your Office 365 directory; unresolved names are surfaced in the reply.

7.2 Conflict checking across all attendees

EMS now runs the Calendar View V2 lookup against **each resolved attendee**. If *any* attendee has an overlap, the slot is treated as busy and the alternative-slot algorithm kicks in. This means the suggestions you get back are free for **everyone**, not just you.

Permissions reminder

Cross-mailbox free/busy lookup requires the Outlook connector account to have Free/Busy or higher permission on each invitee's calendar (the standard Exchange default). If a calendar is unreadable, EMS falls back to checking only the sender and includes a friendly note in the reply.

7.3 Reply email

Both the Booked and Busy templates now render an **Attendees** block via the {AttendeesList} token. Unresolved names appear in italic with a 'needs your confirmation' hint.

8. NEW — Working-Hours-Aware Suggestions

v1.3 makes one promise loud and clear: **EMS will never suggest a meeting outside the working hours you configured**. The suggestion engine now uses *Workday Start Local* and *Workday End Local* as hard boundaries.

8.1 Updated suggestion algorithm

1	Determine the local working window for the requested day (<i>Workday Start Local</i> → <i>Workday End Local</i>) in the resolved time zone from §6.
2	Compute start pointer = max (<i>Workday Start</i> , rounded-up-to-15-min, <i>now</i> + Suggestion Lead Time).
3	Iterate in 15-minute steps. For each candidate, build <i>candidate end</i> = <i>candidate start</i> + DurationMin.
4	Reject any candidate where <i>candidate end</i> exceeds <i>Workday End Local</i> .
5	Convert to UTC and run Calendar View V2 against every attendee.
6	Keep non-overlapping windows; stop after Suggestion Count.
7	If the day yields fewer than Suggestion Count valid slots, automatically extend to the next working day(s) until the quota is filled.

Why this matters

No more 7:30 AM proposals because that was the only 'free' slot. No more 8 PM Friday suggestions. EMS now respects your team's actual working hours, every single time.

9. AI Builder Extraction Schema (v1.3)

Paste the JSON below into the *Extract Schema JSON* EV. The two new fields (**attendees** and **detected_timezone**) power the v1.3 features. **proposed_datetime_text**, **duration_minutes**, **timezone**, and **normalized_local_iso** behave exactly as in v1.2.

```
[
  { "name": "proposed_datetime_text", "type": "string",
    "description": "Return ONLY the exact date/time phrase from the email. Examples: '26 Oct at 2 pm', '11 Oct 15:00'. Do not add words or normalize." },
  { "name": "duration_minutes", "type": "number",
    "description": "Integer minutes if explicit. Accept: 25 min, 45 minutes, half an hour (30), quarter hour (15), 1.5 hours (90), 2 hours (120), 1h 15m (75). Else empty." },
  { "name": "timezone", "type": "string",
    "description": "Explicit TZ ONLY if present (abbrev, IANA, Windows, or numeric). Else empty." },
  { "name": "detected_timezone", "type": "string",
    "description": "NEW v1.3. Best-guess IANA TZ inferred from any TZ cue in the email (PST, CET, UTC+5, +05:00, Asia/Karachi, Pakistan Standard Time). Else empty." },
  { "name": "normalized_local_iso", "type": "string",
    "description": "'YYYY-MM-DDTHH:mm' using the EXACT typed time. noon=12:00, midnight=00:00. Do NOT convert. Year is overwritten by EMS." },
  { "name": "attendees", "type": "array",
    "description": "NEW v1.3. Array of email addresses or display names referenced as participants in the body (excluding sender). Use email when present, otherwise the name as written." }
]
```

10. Core Expressions (reference)

10.1 ResolvedTZ (NEW)

```
coalesce(
  outputs('detected_timezone'),
  outputs('timezone'),
  parameters('Default Time Zone (suhaib_DefaultTimeZone)')
)
```

10.2 StartLocalISO (keep typed time; overwrite year)

```
concat(
  parameters('Default Year (suhaib_Default_Year)'), '-',
  substring(coalesce(outputs('NormISO'), formatDateTime(utcNow(), 'yyyy-MM-dd')), 5, 5),
  'T',
  coalesce(substring(outputs('NormISO'), 11, 5),
    parameters('Workday Start Local (suhaib_WorkdayStartLocal)'),
    ':00'
  )
)
```

10.3 StartUTC_Rolled (UTC start, TZ-aware)

```
convertFromUtc(
  outputs('StartLocalISO'),
  outputs('ResolvedTZ')
)
```

10.4 DurationMin

```
coalesce(
  outputs('duration_minutes'),
  parameters('Default Meeting Duration (suhaib_DefaultMeetingDuration)')
)
```

10.5 EndUTC

```
formatDateTime(  
  addMinutes(outputs('StartUTC_Rolled'), int(outputs('DurationMin'))),  
  'yyyy-MM-ddTHH:mm:ssZ'  
)
```

10.6 Attendees array (NEW)

```
union(  
  triggerOutputs()?['body/toRecipients'],  
  triggerOutputs()?['body/ccRecipients'],  
  outputs('attendees')  
)
```

10.7 Conflict-check array length (per attendee)

```
length(coalesce(  
  body('Get_calendar_view_of_events_(V2)')?['value'],  
  createArray()  
)
```

11. Conflict Semantics

EMS treats two windows as conflicting if any portion overlaps. Outlook's Calendar View V2 returns items with start/end in their own timezone. EMS converts each to UTC and compares ticks across **every attendee**.

If the union of overlap arrays is empty → **book** the event. Otherwise → generate alternative slots inside working hours.

12. Validation Scenarios

A. Happy path (single attendee)

Subject: Meeting request: 26 Oct 2:00 pm — 30m. **Body:** 'Can we meet on 26 Oct at 2:00 pm for 30 minutes?' **Expected:** Event booked; Booked reply sent.

B. Conflict path

Place a placeholder event overlapping 26 Oct 14:00; resend the email. **Expected:** Busy reply with 3 alternatives, all within working hours.

C. Missing duration

Omit duration. **Expected:** EMS uses Default Meeting Duration.

D. Missing year

Omit year. **Expected:** EMS overwrites with Default Year and rolls per Roll Past to Next Year.

E. Explicit time zone in text (NEW)

Include 'CET' or '+05:00'. **Expected:** Reply renders in the detected TZ; calendar invite stored at the correct UTC instant.

F. Multiple attendees in To/Cc (NEW)

Send the request with two extra people on Cc. **Expected:** Event includes all three as required attendees; conflict check runs across all calendars.

G. Attendees mentioned by name in body (NEW)

'Loop in Sara and Bilal please.' **Expected:** Resolved names appear in invite; unresolved names noted in reply.

H. Working-hours boundary (NEW)

Workday End = 18:00, Duration = 60, request at 17:30. **Expected:** Slot rejected for ending past 18:00; alternatives roll to next working day.

I. Lead time respected

Ask for a slot before *now* + *lead time*. **Expected:** First suggestion $\geq$ *now* + *lead time*.

13. Test Matrix (Minimum set)

ID	Case	Input Example	Expected
T1	Book basic	26 Oct 14:00, 30m	Event booked; Booked email.
T2	Conflict alternatives →	Busy at 26 Oct 14:00	Busy email with 3 options inside working hours.
T3	Missing duration	26 Oct 14:00	Booked with Default Duration.
T4	Missing year	10 Oct 10:00 (no year)	Uses Default Year; rolls if needed.
T5	Explicit TZ in text (NEW)	11 Oct 15:00 CET	Reply rendered in CET; UTC stored correctly.
T6	Lead time respected	Ask for < now + lead	First suggestion ≥ lead.
T7	Multi-attendee booking (NEW)	Cc: alice@x, bob@x	All three on the invite; cross-mailbox conflict check.
T8	Named attendees (NEW)	'Loop in Sara'	Sara resolved from directory and added.
T9	Working-hours floor (NEW)	Now = 06:30, Workday Start = 09:00	First suggestion ≥ 09:00.
T10	Working-hours ceiling (NEW)	Workday End = 18:00, 60-min duration at 17:30	Slot rejected; rolls to next day.

14. Troubleshooting

- **Schema mismatch / 'type' not defined:** ensure the Extract Schema JSON matches §9.
- **Wrong year:** verify Default Year and Roll Past to Next Year.
- **Reply shows wrong time zone:** check that *detected_timezone* is being populated. If your AI Builder model is older, retrain or hard-code the schema.
- **Attendees not added:** confirm the Office 365 connector account has Free/Busy permission and that names resolve in the directory.
- **No alternatives returned:** widen the workday window or lower the duration; the strict working-hours rule will reject anything that overflows.
- **AI Builder errors:** confirm credits and Dataverse connection are healthy.

15. Upgrading From v1.2 → v1.3

- Replace the *Extract Schema JSON* EV with the v1.3 schema in §9.
- Add the *ResolvedTZ* compose step (§10.1) before any reply formatting.
- Update *StartUTC_Rolled* to the TZ-aware variant in §10.3.
- Replace the static 'To' in *Create event (V4)* with the union expression in §10.6.
- Add an attendee loop around *Get calendar view of events (V2)* for cross-mailbox conflict checking.
- Update Booked and Busy reply HTML to use {AttendeesList} and {ResolvedTZ} tokens.

16. Support & Cleanup

Uninstall: Turn off EMS flows; remove the solution; delete unused connections.

Support: cs-arpatech.group-1@arpatech.com • Coverage: 24x5.

You're set. Go schedule something great.

EMS v1.3 takes the friction out of cross-team, cross-time-zone scheduling. Spin it up, send a test email, and watch your inbox become a real assistant.