Amtiri

REFERENCE MANUAL

Mailer

Generated on November 25, 2026

The Mailer sends an e-mail when a variable changes: an alarm, a level out of range, a pump that stopped. It sends through a mail server you configure, with your own account, and each message can carry the current values of other variables in its subject and text.

Module	What it does	Typical use
Mail Sender (`MAILER`)	Sends one e-mail per change of a trigger variable, when an enabler allows it	Alarm notices to an operations or maintenance mailbox

The module also has the settings every module shares: whether it is enabled, its name, its status and message variables and their thresholds. They are described in Settings every module shares.

Mail Sender

A Mail Sender holds the mail server's settings once, and a list of **messages**, one per trigger variable. Each change of a trigger variable sends its message, when the message's enabler allows it.

Settings

Key	Setting	Type	Default	Meaning
`smtpHost`	SMTP host	text	none	The mail server's name or address. Required: without it the module does not open
`smtpPort`	SMTP port	int	`587`	The mail server's port. The editor saves `587` when the field is empty or not a number
`smtpSecurity`	Security	`STARTTLS` , `SSL` or `NONE`	`STARTTLS`	How the connection is protected. The editor shows STARTTLS, SSL /TLS and None. Empty means `STARTTLS` ; any other value and the module does not open
`smtpUser`	User	text	empty	The account to sign in with. Empty means the module does not sign in
`smtpPassword`	Password	text	empty	The account's password, written as it is or as `env:NAME` to read it from an environment variable. Used only with `smtpUser`
`fromAddress`	Sender address	text	empty	The address the mail comes from. Empty means `smtpUser`
`variables`	—	map of messages	empty	The messages, one per trigger variable, keyed by its code

- **Security.** `STARTTLS` connects in plain text and switches to TLS with the STARTTLS command, usually on port 587. It is required: a server that does not offer STARTTLS gets no mail, and the send fails. `SSL` uses TLS from the first byte (SMTPS, implicit TLS), usually on port 465. `NONE` sends everything, the password included, in plain text: use it only for a relay inside a network you trust.

- **The server's certificate.** With `STARTTLS` and `SSL`, the certificate the server presents must name the host written in `smtpHost`, and be trusted by the Java runtime that runs the engine. Write the name the certificate carries, not an IP address it does not list.
- **No account.** With `smtpUser` empty the module sends without signing in, which suits a relay that accepts mail from the plant's network. `fromAddress` must then be set: with both empty, the module does not open.

> **Important**
>
> Write the password as `env:NAME`, for example `env:DO_SMTP_PASSWORD`, rather than the password itself. The module record is a file that is copied, exported and transferred with the configuration, and `env:NAME` keeps only the variable's name in it. The environment variable must be set for the service that runs the engine, such as the `Environment=` line of a systemd unit or the environment of a container: a variable set in your own shell is not seen by the service. The password is read each time the module opens, and a variable that is not set gives an empty password: the server refuses it, and the module shows `The mail server refused the credentials`.

Messages to send

Each message is an entry of `variables`. The editor lists them in a table, with the sender, subject and body in a dialog opened from the row's pencil. A Mailer saved with a Config editor older than 3.13.0 holds its messages in a form the engine cannot read, and does not load.

Key	Setting	Type	Default	Meaning
`variable`	Variable	variable code	none	The trigger: each change of its value sends this message. One message per variable
`enabler`	Enabler	expression	empty	An Amtiri Script expression that must give `true` for the message to be sent. Empty means always
`destinations`	Recipients	text	none	The recipients' addresses, separated by commas
`sender`	Sender	text	empty	The name the recipients see beside the sender address
`subject`	Subject	text	empty	The subject line. `{CODE}` placeholders are filled in
`body`	Body	text	empty	The text of the mail, sent as plain text. `{CODE}` placeholders are filled in

Behaviour

- **Every change is a trigger.** A change of the trigger variable in either direction sends the message, when the enabler allows it. An alarm that comes and goes sends two mails unless the enabler lets only one through.
- **Placeholders.** A variable's code in braces, such as `{LIT_301}`, is replaced in the subject and the body by that variable's current value. A variable that holds no value leaves the placeholder empty. A code in braces that names no variable is left as typed, braces included.
- **The mail says what the plant looked like at the change.** The enabler and the placeholders are evaluated when the change is handled. The finished mail is then queued.
- **Mail goes out in order, off the formulas' path.** Queued mails are sent one at a time, in the order they were triggered, on the module's own thread, so a slow mail server never holds up formulas or other modules. Mails still queued when the module stops are sent.
- **What a mail carries.** The sender address, with the message's sender name beside it, the recipients in To, the subject, and the body as plain text, all in UTF-8. There is no Reply-To, Cc or attachment.
- **Timeouts.** Connecting to the mail server and waiting for its answers each time out after 10 seconds.
- **Failures.** A server that refuses the account is an error at once, since trying again cannot help. Any other failed send is counted, and so is an enabler or a placeholder that fails; the next mail that goes through clears it.

> **Warning**
>
> Choose the trigger with care. A measured value that changes every few seconds, used as a trigger, sends a mail every few seconds. Trigger on an alarm variable instead, and keep the measurement in a placeholder.

Messages

A counted failure is a `WARNING` that reads `… (1 of 3)`, `… (2 of 3)` and becomes an `ERROR` reading `… after 3 attempts`, with the thresholds set in Settings every module shares. The engine's own messages are in Module messages.

When	Level	Text
Opening: no SMTP host is set	`ERROR`	`No SMTP host`
Opening: neither a sender address nor a user is set	`ERROR`	`No sender address`
Opening: `smtpSecurity` is not `STARTTLS`, `SSL` or `NONE`	`ERROR`	`Unsupported SMTP security '{security}'`
Sending: the mail server refuses the user or password	`ERROR`	`The mail server refused the credentials`
Sending: the mail could not be delivered	`WARNING`, then `ERROR`	`Mail delivery failed: {reason}`

Example

A high-level alarm on tank 301, sent to two mailboxes through an authenticated server:

```
SMTP host        smtp.example.com
SMTP port        587
Security         STARTTLS
User             plant-alerts@example.com
Password         env:DO_SMTP_PASSWORD
Sender address   (empty: plant-alerts@example.com)

Variable     LIT_301_HH
Enabler      LIT_301_HH
Recipients   operations@example.com, maintenance@example.com
Sender       Water plant alarms
Subject      High level in tank 301: {LIT_301} m
Body         Tank 301 reached {LIT_301} m. Pump P-301 running: {P_301_RUN}.
```

`LIT_301_HH` is a boolean alarm. When it turns `true`, the enabler lets the message through, and the mail carries the level and the pump's state. When it turns back to `false`, the enabler gives `false` and nothing is sent.

Next steps

- Settings every module shares
- Module messages
- Module catalogue

This page describes Data Orchester engine 6.12.0.