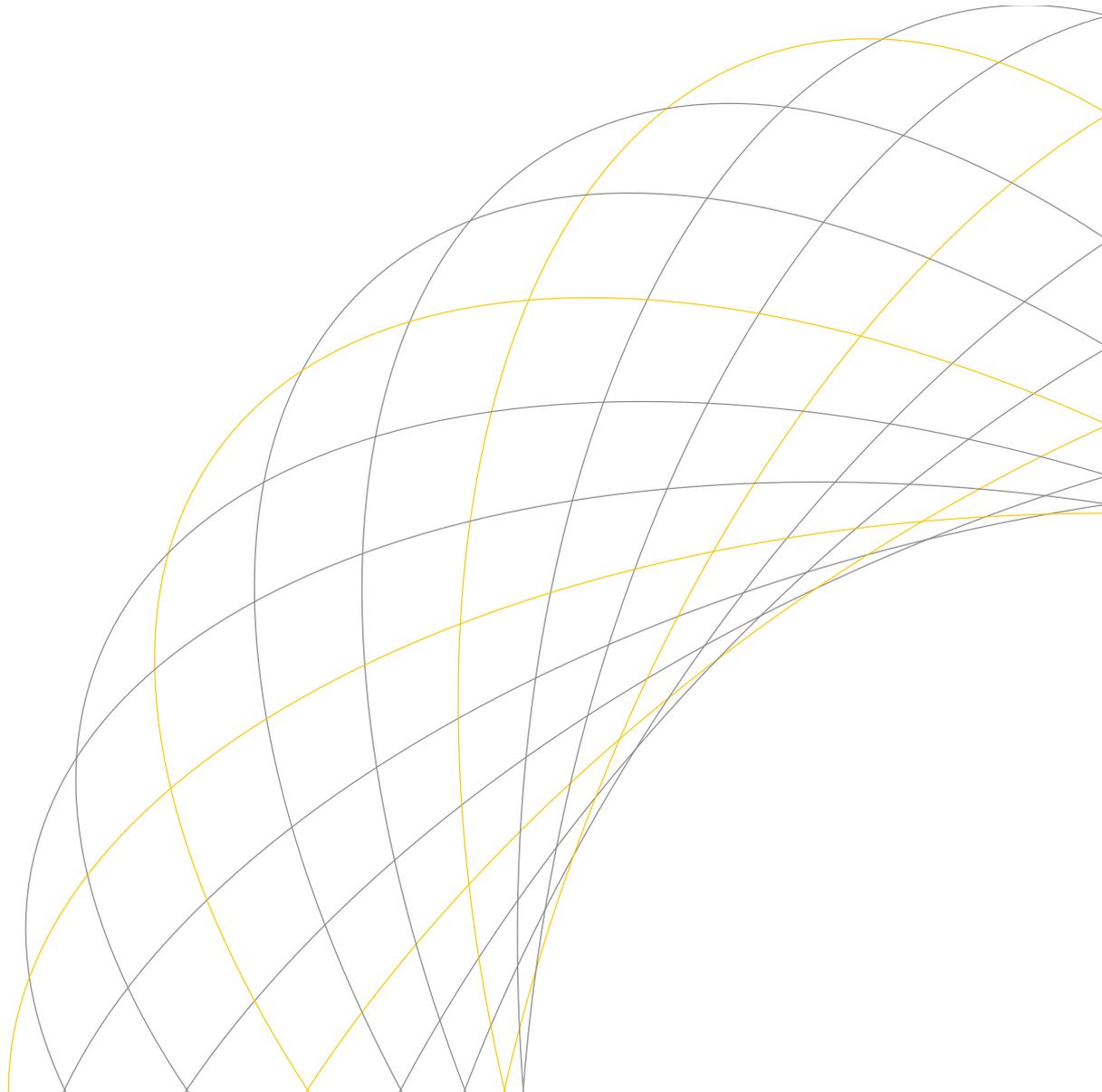




# Mail Integration Solution Guide



Contents

**GENERAL INFORMATION..... 3**

**About IMAP ..... 3**

**About Rapid Automation (RA) ..... 4**

    RA Agent ..... 4

    RA Solution..... 4

**About Mail Integration Solution ..... 5**

**ADMINISTRATION ..... 6**

**Requirements ..... 6**

**Installation ..... 6**

**Configuring system objects..... 7**

    Agent object..... 7

    Connection object..... 8

**WORKING WITH JOBS ..... 9**

**Creating jobs..... 9**

    Common definition - example and parameters..... 10

    DOWNLOAD definition- example and parameters ..... 12

    COPY definition- example and parameters ..... 14

    DELETE definition- example and parameters ..... 15

    Field names ..... 16

**Executing jobs..... 17**

**Return Codes ..... 18**

**Error handling..... 19**

## General information

The Rapid Automation (RA) Solution integration allows the *Automic Automation Platform* to access any IMAP mailbox and check for incoming emails.

### About IMAP

The Internet Message Access Protocol (commonly known as IMAP) is an Application Layer Internet protocol that allows an email client to access email on a remote mail server. The current version, IMAP version 4 revision 1, is defined by RFC 3501. An IMAP server typically listens on well-known port 143. IMAP over SSL (IMAPS) is assigned well-known port number 993.

IMAP supports both on-line and off-line modes of operation. Email clients using IMAP generally leave messages on the server until the user explicitly deletes them. This and other characteristics of IMAP operation allow multiple clients to manage the same mailbox. Most email clients support IMAP in addition to Post Office Protocol (POP) to retrieve messages; however, fewer email services support IMAP. This protocol offers access to the mail storage. Clients may store local copies of the messages, but these are considered to be a temporary cache.

Incoming email messages are sent to an email server that stores messages in the recipient's email box. The user retrieves the messages with an email client that uses one of a number of email retrieval protocols. Some clients and servers preferentially use vendor-specific, proprietary protocols, but most support IMAP, SMTP for sending email and POP and IMAP for retrieving email, allowing interoperability with other servers and clients.

For example, Microsoft's Outlook client uses MAPI, a Microsoft proprietary protocol, with a Microsoft Exchange Server. IBM's Notes client works in a similar fashion while communicating with a Domino server. All of these products also support POP, IMAP, and outgoing SMTP. Support for the Internet standard protocols allows many email clients such as Pegasus Mail or Mozilla Thunderbird to access these servers, and allows the clients to be used with other servers.

## **About Rapid Automation (RA)**

Rapid Automation is a generic technology that can integrate various solutions; as a result, it connects to particular external applications and platforms.

### **RA Agent**

The RA Agent provides the universal framework and connects the Solution with the *Automic Automation Platform*.

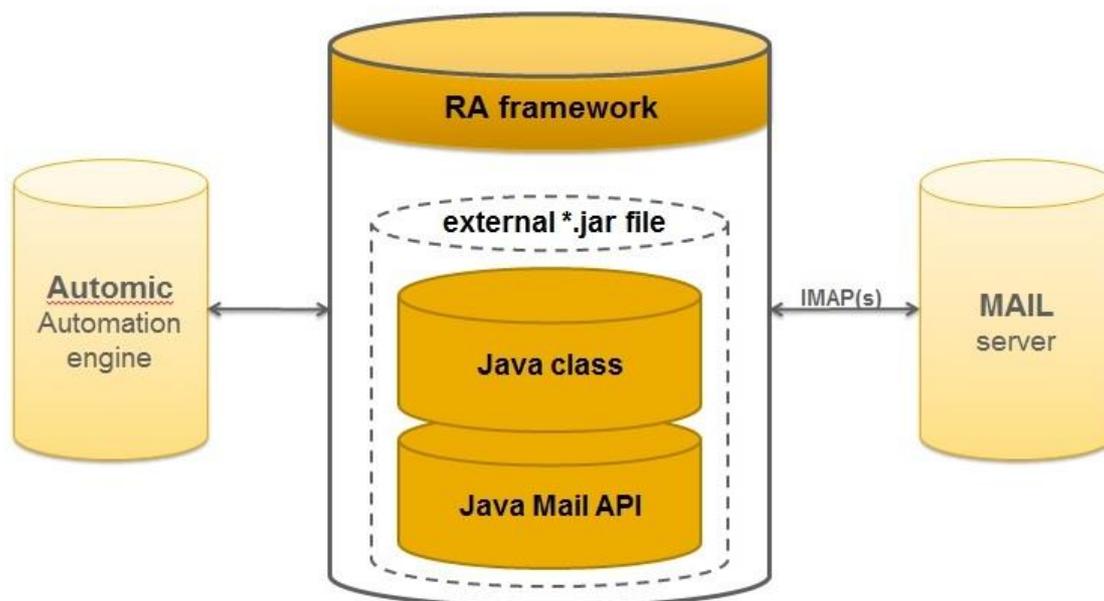
The Key benefit of this concept is that the same basic agent can be combined with any other Solution of your choice. That is why a lot of different RA Agents are in place and delivered by *Automic Software*.

### **RA Solution**

The RA Solution is linked with an RA Agent. It determines the Agent's functionality, and the RA Connection objects, Agent objects and Job objects which can be created and how they are structured. The Solution is supplied as a JAR file.

## About Mail Integration Solution

The following illustration describes the main architecture of the Rapid Automation **Mail Integration Solution**. As already mentioned, the **Mail Integration Solution** was initially developed as a standalone Java class using a Java Mail API provided by Oracle (formerly Sun).



This pre-developed Solution was built and packaged as an external JAR file and used within the Rapid Automation framework. The final result represents the RA **Mail Integration Solution** which can be easily integrated into the *Automic Automation Platform*.

By using the Mail Agent class and any external IMAP API, the agent is able to connect to any IMAP mailbox. Additionally, the IMAPS protocol can be used to satisfy highest security policies. Finally, the agent will process incoming emails to a specified system folder on the agent's host. Downloaded emails will be marked as "read". Attachments can also be downloaded with the **Mail Integration Solution**.

Therefore it represents an interface which can be used as a trigger to control the *Automic Automation Platform* by incoming email.

## Administration

This section describes the installation of all required components for the RA *Mail Integration Solution*.

### Requirements

To ensure a smooth operation of the *Mail Integration Solution*, the following requirements have be taken into account:

- Automation Engine (at least Version 8, because of RA technology)
- Rapid Automation framework (at least version 6.4.0, at least 4.1.0 for AE Version 8)
- Java Runtime Environment (at least version 7)

### Installation

This describes the installation process of an RA Agent.

Install the Agent for Rapid Automation (installation image / DVD under: \Agents\rapid-automation\Core\) as described in the *Automic Automation Platform* documentation. This documentation also describes how the RA Solution is loaded into the Automation Engine database.

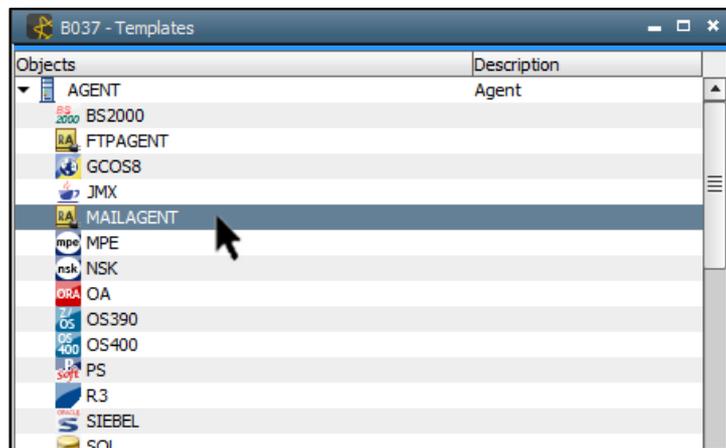
Please keep in mind that (for this RA Solution as well as others) you first have to create an Agent before you can execute the Agent's process.

## Configuring system objects

The following shows how to configure necessary objects of the RA Solution.

### Agent object

To create a new Mail Agent before starting the Mail Agent's process, please login to client 0. Preferably use folder "HOSTS" to create a new agent object.



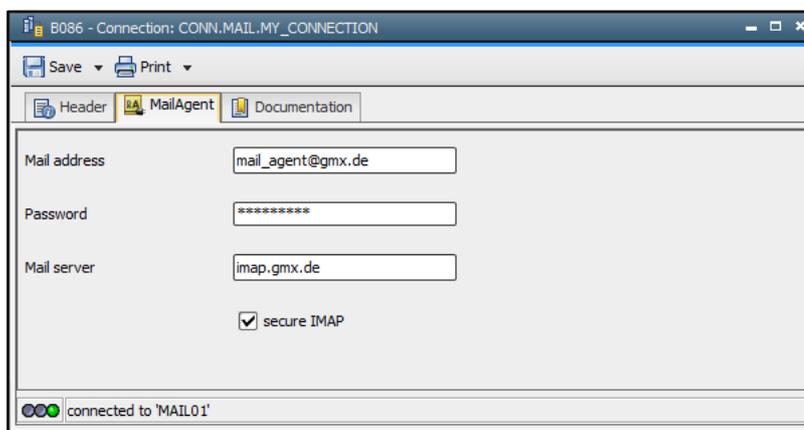
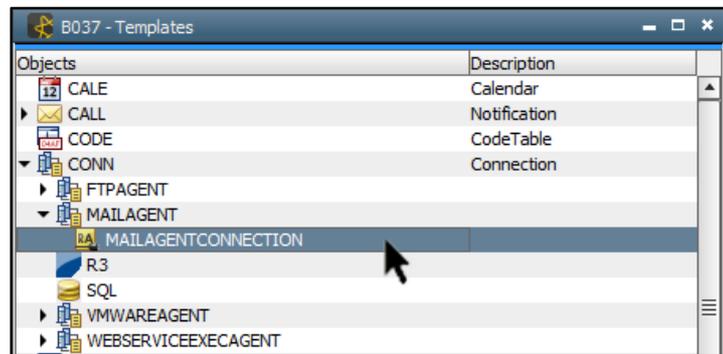
As described in the *Automic Automation Platform* documentation, name the Agent as it was configured within the INI file of the Agent's installation.

After executing the Agent's JAR file, the Agent is started and should be available in the system.

Further configuration on Agent's level is not necessary. Only connection objects for the desired mailboxes have to be created.

## Connection object

Similar to the Login objects, create one or more connection objects in the desired client.



These connections are used within the Job's object definition and include the host (Mail server), user (Mail address), password and desired IMAP protocol to be used for performing the job.

By using more than one connection object, multiple mailboxes are accessible. A combination of multiple mail server addresses within one connection object is not possible.

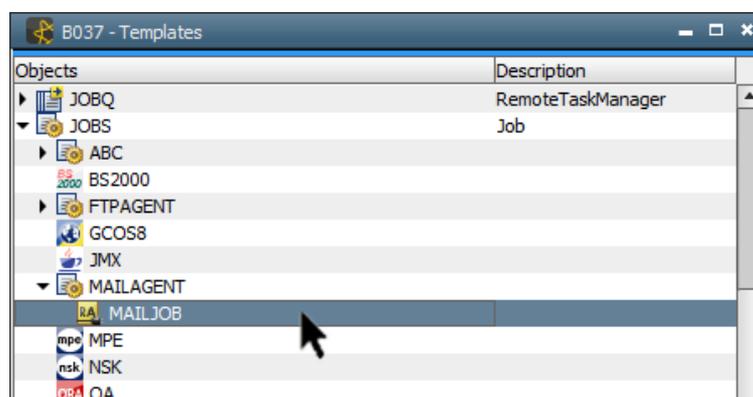
Finally, after starting the agent and creating at least one single connection object, all prerequisites for the configuration of the Mail Agent are done.

## Working with Jobs

This topic should illustrate how the **Mail Integration Solution** is working and what kind of options users have when using Mail Agent jobs.

### Creating jobs

To create a new Mail Agent job, create a new object as usual. The template list will offer a single Mail Agent entry called "MAILJOB". This is the main Mail Agent job template to be selected.



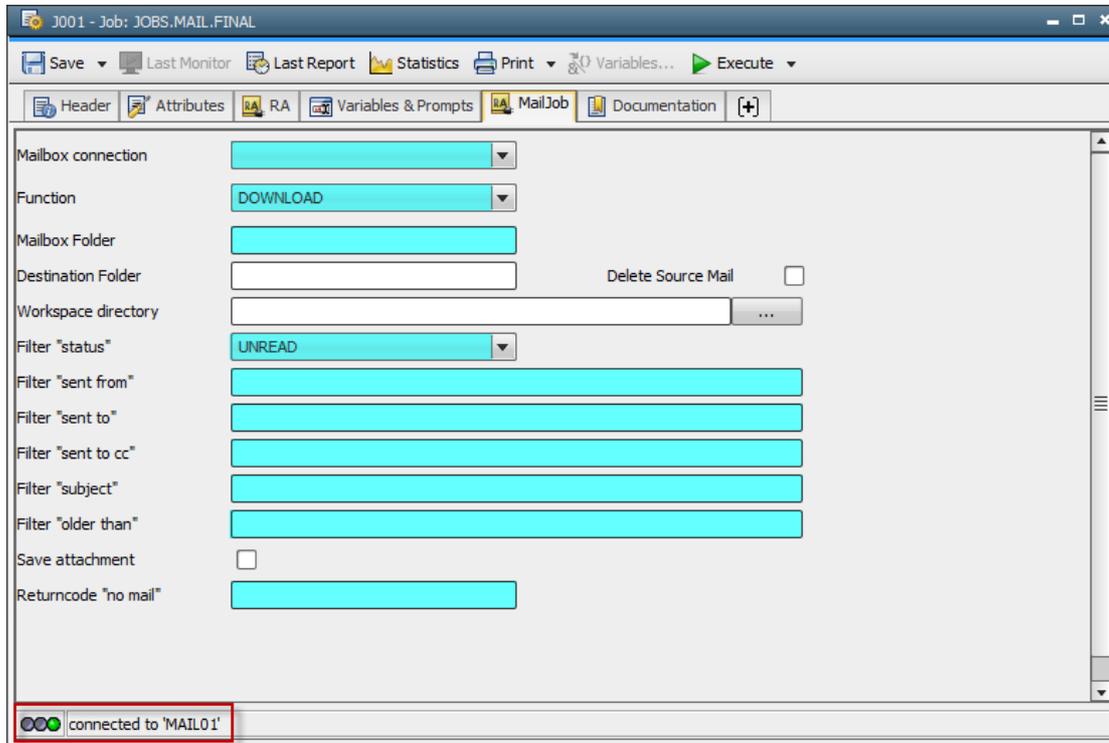
After selection of the Mail Agent job template, the object should be named as desired but, as usual, taking your object naming conventions into account.

After creating the new Job object, the following modifications have to be done to finalize the object definition.

Within the Attributes tab, the Mail Agent object has to be specified if available. If not, please ensure that the Agent is up and running and authorized (read, write and execute) for this client. Especially in case of missing authorizations the agent will not be visible.

## Common definition - example and parameters

The following screenshot illustrates the Mail Agent's default main tab (called *MailJob*). You will first get this "empty" screen after you have just defined the job object and entered the Mail Agent's name in the Attributes tab (the red frame at the bottom marks the area showing if your Mail Agent – in this case "MAIL01" - is up and available):



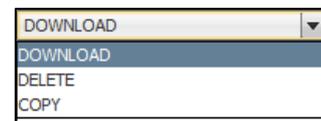
The fields highlighted in blue color are those which are always available, no matter which function you select. These will be explained first, upside down.

### Mailbox Connection

First, choose the desired connection object which includes the connection credentials to the targeted mailbox. If the connection object is not available, create one within the same client.

### Function

This field defines the desired functionality. By processing the mailbox, all included mails can be downloaded (DOWNLOAD), copied to another mailbox folder (COPY) or deleted (DELETE).



After changing the drop down entry, only the relevant fields for the corresponding function will remain on your display.

### Mailbox folder

By default, the mailbox folder which has to be processed most is the default INBOX folder. Nevertheless it is possible to choose a different mailbox folder, like a subfolder within the mailbox folder tree (Format: <FOLDER>/SUBFOLDER/>...).

## Filter

Before downloading mails, filters can be used to include or exclude mails from processing.

The desired mail **Status** (UNREAD, READ, ALL) can be selected from a dropdown menu, the default is: UNREAD



Using regular expressions, the following mail fields can be parsed additionally:

- **Sent from** (filter using regular expressions)
- **Sent to** (filter using regular expressions)
- **Sent to CC** (filter using regular expressions)
- **Subject** (filter using regular expressions)
- **Older than** (amount of days since mail was received)

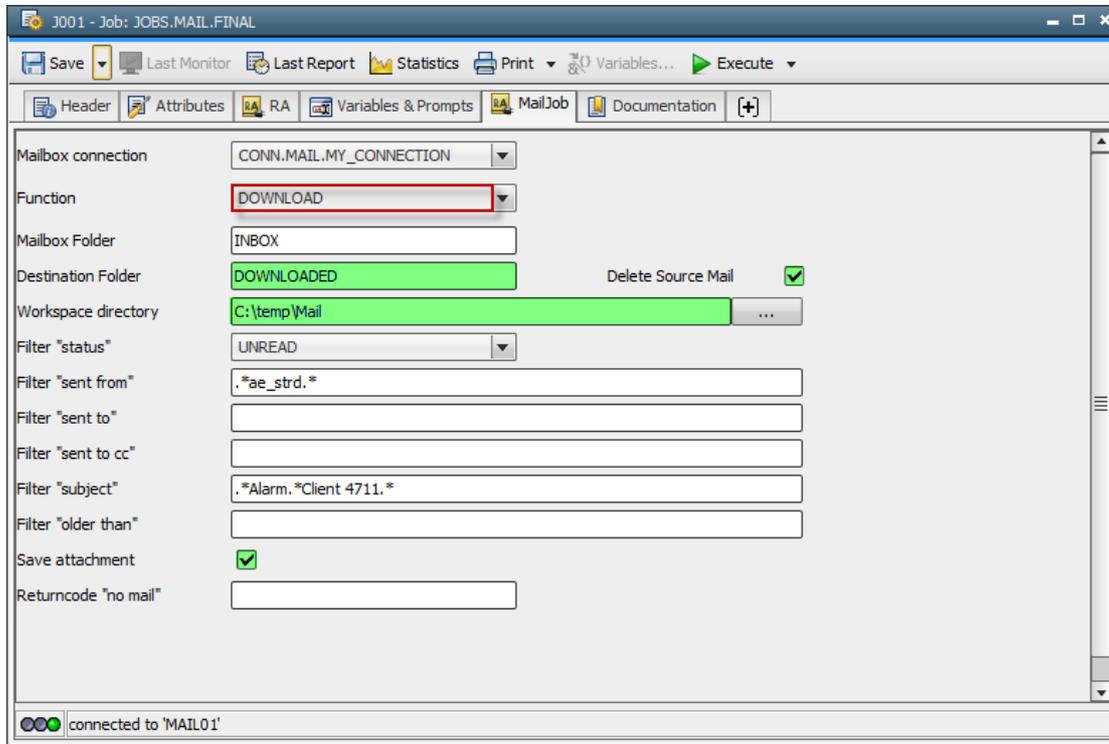
Within the later screenshots and parameter descriptions for the different functions we will also show a few examples of most commonly used regular expressions for filtering.

### Returncode “no mail”

Here, you can define a numeric return code which will be returned if no mail was selected. If unchanged, “no mail” will return “1”.

## DOWNLOAD definition- example and parameters

In the following screenshot, the specific DOWNLOAD parameters are highlighted in green color. We will list and explain them in this section.



This job will look at the INBOX folder of the mailbox which has been specified within the connection object "CONN.MAIL.MY\_CONNECTION". As you can see in the screenshot, this DOWNLOAD definition will filter for UNREAD mails only, containing string "ae\_strd" somewhere within the "sent from" field (part or all of the sender's name); additionally, the mails from this sender must contain the strings "Alarm" and "Client 4711" within the mail subject field (again, the position of both strings within the subject does not play any role, but the sequence is important here).

So far, the used fields already have been explained in the "Common" section; now let us have a look at the specific DOWNLOAD parameters:

### Workspace directory

The workspace directory represents the folder on the Mail Agent machine where all mails should be stored after download. Attachments will also be stored here (see next parameter). Any mail will be stored within a separate subfolder, named with a timestamp. After download, the selected mails are flagged as "read" in the Mailbox Folder.

### Save attachment

With this flag set (default), not only the selected mails are downloaded to the workspace, but also any attachments to these mails.

The following parameters enhance the DOWNLOAD function and provide the ability to combine the mail and attachment DOWNLOAD with COPY and DELETE functions:

**Destination Folder**

If you enter a mailbox folder name here which is different from the Mailbox Folder defined in the correspondent field above, the original mail will not only be flagged as “read” (if selected), but also copied to this mailbox folder. Keep in mind that this folder must exist within the selected mailbox.

**Delete Source Mail**

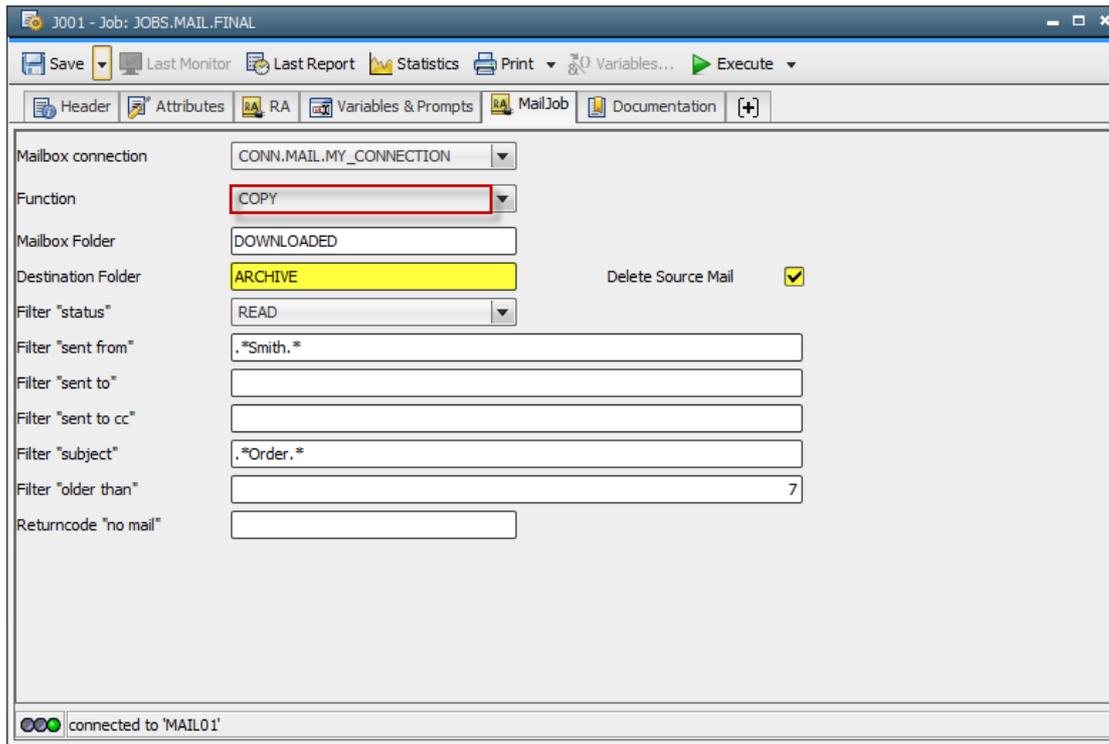
This flag is for the purpose to not only copy the selected mails to the specified Destination Folder, but to move it (copy + delete original mail) instead.

Additionally, you can also use it to just delete the source mail after it has been downloaded. For this purpose, simply set this flag and leave the Destination Folder field empty.

## COPY definition- example and parameters

Although we already explained how to copy or move mails with the DOWNLOAD function, additionally to the download itself, there may be situations where you just need to copy or move mails from one mailbox folder to another without the need to have a downloaded copy available. This is where the COPY function comes in.

The following screenshot shows a COPY definition example, the specific COPY parameters are highlighted in yellow:



Similar to the DOWNLOAD example earlier, the job above will connect to the same mailbox (see Mailbox connection). But this time, it will look into the mailbox folder DOWNLOADED instead of INBOX, for any mail already read (status) which was sent by any “Smith” (sent from), using the string “Order” within the subject. But only if they are older than 7 days (older than).

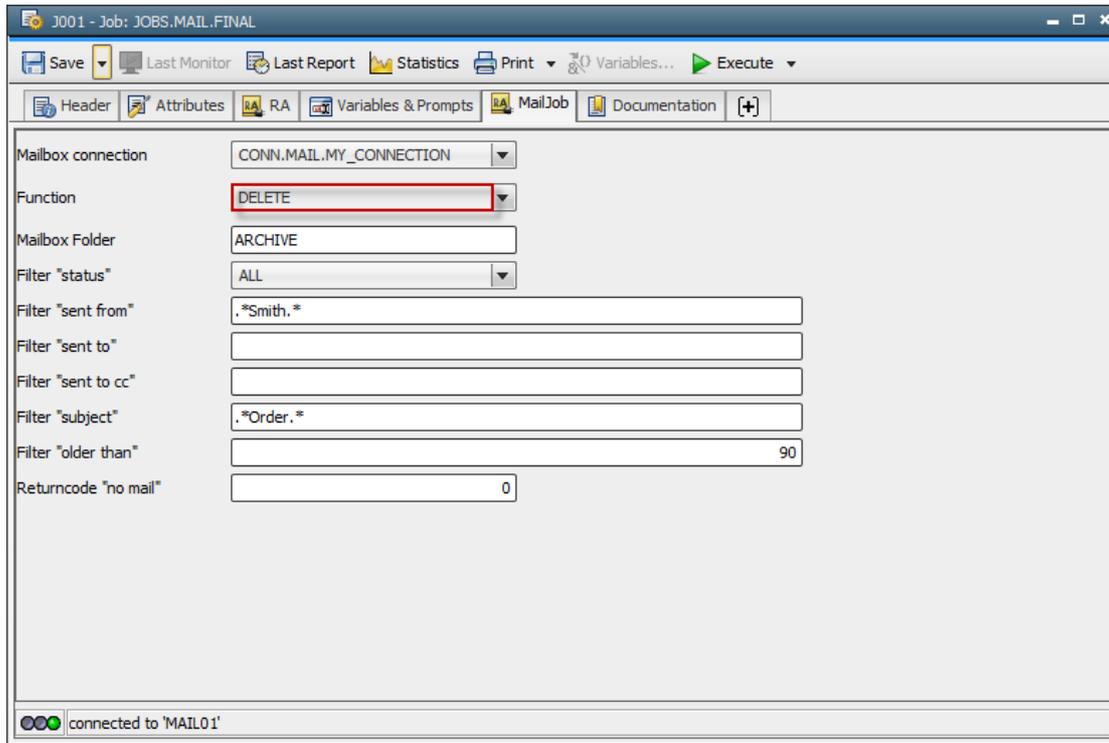
Another difference to the example before: the COPY function needs no “Workspace directory” and also no “Save attachment” field as nothing is downloaded from the mailbox.

But a “Must” for a COPY function is a destination (in this case ARCHIVE) where to copy or move (Delete Source Mail flag) the original mail to. So these are the parameters specific to the COPY function. As they work exactly like they do with the DOWNLOAD function (see above), there is no need to explain them again.

The screenshot above shows a simple housekeeping job for a specific mailbox: take all “Order” emails older than 7 days from sender(s) “Smith” which have already been processed somehow (already read and moved to folder DOWNLOADED), copy them to the ARCHIVE folder within the same mailbox and delete the copies from folder DOWNLOADED.

## DELETE definition- example and parameters

In order to complete the description of the different **Mail Integration Solution** functions and parameters, we now come to the DELETE function. As you can see in the screenshot below, there are no parameter fields highlighted with any color; only the function itself is marked by a red border. The reason: we already described all available parameters and the DELETE function only uses a subset of the parameters available for DOWNLOAD and COPY.



Similar to the COPY function described earlier, the DELETE function allows to access a mailbox and change their contents, without any further actions like downloading. In order to do so, we need to know where to search (Mailbox connection and Mailbox Folder), what to search for (the different filters) and what to do with the selected mails (Function).

The screenshot above shows a mail job prepared to delete mails from the same mailbox as before (login data defined within CONN.MAIL.MY\_CONNECTION). This time, the job has to select mails from mailbox folder ARCHIVE (the folder we already used before to copy processed orders from sender "Smith" with an age of 7 days or more to). Furthermore, we do not want to differentiate between "unread" or "read" any more and therefore chose status ALL; we also want to finally clear all of these mails which are older than 90 days.

In this example, we also decided to set the return code for "no mail" to zero (0). Why this?!?

By default, "no mail" returns a value of "1", as the given functions cannot be performed if no mail was selected - which may need to be recognized by any following task(s).

In case of our DELETE example it does not make any difference for the following tasks if "selected" mails are not available after the job run because none of them was found or because they were deleted by the mail job. Therefore, we set the return code to zero (0) which will set the mail job status to ENDED\_OK, even if no mail was selected for deletion. Any other error will still produce a return code and cause the job to abort.

## Field names

To ensure that script functions can be used in order to fill the different parameters dynamically at runtime, the different **Mail** fields have to be accessible.

The following table lists all relevant fields and their **Fieldnames** used as attributes:

Display Name	Fieldname	Datatype	Valid values for PUT_ATT
Mail Connection	CONNECTIONV	String	"<Existing Connection Object name">
Function	FUNCTIONV	String	"<DOWNLOAD/DELETE/COPY>"
Mailbox Folder	MAILBOXFOLDERV	String	"<Existing Mailbox Folder>"
Destination Folder	MOVETOFOLDERV	String	"<Existing Mailbox Folder>"
Delete Source Mail	SOURCEDELETEV	Boolean	"<Y/N>"
Workspace Directory	WORKSPACEV	String	"<Existing agent directory>"
Filter "status"	FILTERSTATUSV	String	"<ALL/READ/UNREAD>"
Filter "sent from"	FILTERSENTFROMV	String	"<Regular expression>"
Filter "sent to"	FILTERSENTTOV	String	"<Regular expression>"
Filter "sent to CC"	FILTERSENTTOCCV	String	"<Regular expression>"
Filter "subject"	FILTERSUBJECTV	String	"<Regular expression>"
Filter "older than"	FILTEROLDERTHANV	Integer	"<Number of days (1-n)>"
Save attachment	SAVEATTACHMENTV	Boolean	"<Y/N>"
Returncode "no mail"	RETURNCODENOMAILV	Integer	"<Numeric value (0-n)>"

### Explanation:

- **"..."** - Always include input values in quotes (both single or both double)
- **<...>** - Replace brackets and included text by real value
- **/** - Separates valid input values

Example: When you read "**<Y/N>**", you may write it as "**Y**", "**Y'**", "**N**" or "**N'**" instead, where **Y** means **Yes** and **N** means **No**.

When using script functions like PUT\_ATT in the Pre-Process tab or Process tab of the Mail job, those fields are accessible at runtime. This allows more flexibility in creating and designing generic jobs.

## Executing jobs

This section describes and explains the processing of each job function during execution. The main differences of the various functions are shown here:

- Connect to the mail account
- Iterate all available emails
- Include or exclude emails to be processed by using predefined filter definitions
- Function: **DOWNLOAD**
  - Download all included emails to the defined working directory
  - Emails are stored in separate folders by using a unique timestamp (Format: System timestamp in milliseconds)
  - Creating memo TXT file (containing all values from the original email like: body, subject, sent to, sent from,...); can be used for follow-up processing
  - Download attachments (PDF, TXT, etc.); attached emails are stored as files using the following format: <SUBJECT>.eml. Special characters are replaced by “\_”
  - Finish file (after complete and successful download of the email an empty finish indicator file “xxx\_finish.txt” is created); can be used for follow-up processing as well
  - Email is marked as read
  - If user defined a Destination Folder → email will be copied or moved there, depending on the “Delete Source mail” flag
- Function: **COPY**
  - Email is marked as read
  - Email will be copied or moved to the Destination Folder, depending on the “Delete Source mail” flag
- Function: **DELETE**
  - Delete all selected emails, without any checkback!
- Disconnect from mail account

## Return Codes

The following should describe all upcoming return codes:

Name	Description
0	No errors; at least one email was selected and processed
1	No email was found / No emails matched the defined filters Can be changed by parameter <b>Returncode "no mail"</b>
2	At least one email was selected Final move to the Destination Folder did not work (Please check if the folder exists or if the folder is available via IMAP!) <b>NOTE: This return code indicates errors with the final email move only</b>
>2	Unique error codes Connection issues Access problems Encoding errors ...

## Error handling

The following can be used to check issues with the ***Mail Integration Solution***:

- Use Telnet to check ports of the mailbox (Firewall issues, IMAP enabled on server level or on account level, etc.)
- Use freeware email client (e.g. Thunderbird) to check the IMAP connection using identical connection data
- Use command line program representing the base of Mail Agent's functionality (please ask *Automic Consulting* for delivery)