Datacap

[Document Capture for a Demanding World]

Taskmaster Email Actions Guide

July 2010

Datacap Inc.
660 White Plains Road
Tarrytown, NY 10591

Copyright Notice

Printed in the United States of America

Trademarks

Datacap Inc.

660 White Plains Road
Tarrytown, New York 10591
Tel: 914-366-0100
Fax: 914-366-0500

Email (Information): info@datacap.com
Email (Support): support@datacap.com
Internet: http://www.datacap.com

Revised July 19, 2010.

Table of Contents

Introduction to Taskmaster Email Actions

The Datacap Taskmaster Email Actions allow you to create Taskmaster document batches from documents received as email attachments, and to send email notification messages when a particular event occurs.

Three actions libraries are included in the Taskmaster Email Actions option:

- IMAP Email Input Actions (IMail.RRX)
- Exchange Web Service Email Input Actions (EWSMail.RRX)
- Email Sending Actions (Email.RRX)

Email Input Actions

Input Actions scan an email inbox for incoming mail messages and place selected messages and attachments into a new Taskmaster batch for processing.

Taskmaster supports two methods of accessing a mail server to obtain the image attachments:

1. The actions in IMail.RRX use the Internet Message Access Protocol (IMAP) for Microsoft Exchange Server, Novell GroupWise, and other mail servers providing support for IMAP.
2. The actions in EWSMail.RRX use Exchange Web Services (EWS) for Microsoft Exchange Server. This is a Simple Object Access Protocol (SOAP) based method of communication.

The Input Actions are typically assigned to a Taskmaster task that is executed by an unattended Taskmaster Rulerunner station. These Input Actions scan one or more inboxes, and allow you to define the type of attachments to include in the batch, for example only TIFF images or only PDF files, which helps eliminate input of improper files.

The batch is created when the maximum number of documents has been received from the email server(s), or a time interval has passed. At the batch level, the EmailCount is captured. Each document in the batch is associated with one email message, and contains fields for the Subject, From, To, DateSent, Priority, and Body. In addition, the IMail actions capture User, and the EWSMail actions capture DateReceived.

When an email is successfully processed and the attachment is input into the Taskmaster batch, the email is moved from the inbox to a specified email “done” folder. If the attachment is not one of the expected types (such as a TIFF file), or there is a problem with processing it, the email and attachment do not become part of the batch and are moved to a specified email “problem” folder.

Note that further processing of the body of an email, or an attachment other than .TIF, after it has been successfully added to a Taskmaster batch using the Email Actions may require you to construct additional rules and to license other Taskmaster options.

To capture MS Office document attachments, .zip files, etc, and turn them into images that can be processed using Taskmaster's standard Recognition or Verify tasks, you can license the eDocument Conversion Actions and put them in your application's processing rules.

If you capture non-image files using the Email Input Actions, the "pages" that are created have non-image files attached to them. If you capture the email body only, the page may not have any files attached. In these cases, your application must handle these documents/pages. If you

try to process a batch containing pages with no image attached using the standard recognition actions or using the verify task, it may not work as you expect.

Email Send Actions

The Send Actions in Email.RRX direct a Taskmaster task to compose and send informational emails under conditions you define. These notification emails can be sent to multiple recipients and can use CC and BCC. The subject line can be specified and can be sent with or without attachments. This is useful for situations where actions encounter exceptions and need to notify a user. For example, if an error occurs during data verification, you can alert an administrator so that he/she can take the appropriate action; the items that caused the failure can be attached to the email. Or, if export rules encounter a problem and a batch is not exported successfully, an email can be composed and sent that contains the details.

Introduction to Taskmaster Email Actions Guide

The purpose of this Taskmaster Email Actions Guide is to provide you with the information you need to install the Taskmaster Email Actions in a Taskmaster 7.5 SP1 or later environment. It also includes reference information on the individual Taskmaster Email Actions that will help you incorporate the actions in your applications.

This Taskmaster Email Actions Guide assumes you have access to a Workstation, or to a client/server environment in which Taskmaster 7.5 SP1 or later is installed and running successfully.

For complete information on installing and configuring the appropriate Taskmaster environment, see the Taskmaster Installation & Configuration Guide found in the root folder on the installation media.

Before You Begin

For information on hardware and operating system requirements for Datacap Taskmaster software, see:

http://dnld.datacap.com/TechNotes/hardware.htm

There are both Taskmaster and mail server prerequisites for using the Taskmaster Email Actions.

Taskmaster Prerequisites

Before you begin installing the Taskmaster Email Actions, ensure:

- Your Taskmaster 7.5 SP1 or later installation is completed and working. At a minimum, ensure one of the sample or foundation applications runs correctly, as this makes troubleshooting the Taskmaster Email Actions installation easier, and narrows down potential problems that may occur.
- You have the Taskmaster Email Actions installation media.
- You have administrator access to all machines on which Taskmaster software is installed.
- When using Email Input Actions, we recommend that you set up an inbox that is dedicated to the emails that are to be processed. If there are pre-existing messages in the Inbox, the earliest messages will be processed first.

Install this option only on machines on which Taskmaster 7.5 SP1 or later software was previously installed. To determine if your software is the correct version on both Taskmaster Client Workstations (Workstations) and Taskmaster Server (TMServer), do the following:

1 From the Windows Start menu, select Settings, then Control Panel. The Control Panel window opens.

2 Double-click Add or Remove Programs to open the Add or Remove Programs window.

3 Ensure the Show updates option is selected.

4 Scroll down to the Datacap Taskmaster entry.

If Datacap Taskmaster 7.5 SP1 or later is not displayed, you cannot install the Taskmaster Email Actions, and you should contact Datacap Support for assistance in obtaining the software upgrades you need.

Email Input Actions Prerequisites

In order to use the Taskmaster Email Input Actions, ensure:

1 You have access to a mail server with one of the following:

- Exchange Web Service (EWS) enabled on Microsoft Exchange 2007 SP1 or later; Datacap recommends that SSL be configured. Use the input actions in EWSMail.RRX.
- Email server supports IMAP protocol and the mail server and firewall (if any) have IMAP access enabled, SSL is not supported at this time. Use the input actions in IMail.RRX.

2 You have an email account you can use for which you know the server URL, and login credentials (user name and password).

3 The dedicated email account contains an Inbox folder and separate folders for messages successfully imported (Done folder) and messages that encountered errors (Problem folder). Note that the names of these folders can be specified using the im_done_folder and im_problem_folder, or ex_done_folder and ex_problem_folder actions.

Email Sending Actions Prerequisites

In order to use the Taskmaster Email Sending Actions (use the actions in Email.RRX), ensure:

1 You have access to an SMTP server that can relay the emails created by the Taskmaster application.

2 The Email Sending Actions are run on a Workstation on which the Windows CDOSYS object or Microsoft Outlook object are registered. The Email Sending Actions use the CDOSYS object by default to send email. If the CDOSYS object is not available, the actions can use the Outlook object. The CDOSYS object is available under Windows XP, Windows 2000, Windows 2003 Server or later.

3 The Email Sending Actions are run on a Workstation and under a user account that has permission to relay or send emails.

Install and Incorporate Taskmaster Email Actions

This section provides the information you need to install and incorporate your Taskmaster Email Actions in two configurations:

- Workstation/Mail Server
  In this configuration, the Workstation hosts all of the Taskmaster components, while a separate email server hosts your IMAP or EWS, and SMTP email software.
- Client/TMServer/Mail Server
  In this configuration:
  - The email server hosts your IMAP, EWS, or SMTP mail software.
  - The TMServer hosts the Taskmaster application at a minimum, and optionally may also host Taskmaster Server, the batch files, databases, and Taskmaster Web.
  - The Workstation hosts the Taskmaster Client software (Taskmaster Client and/or Datacap Studio).
  - Other unattended stations host Taskmaster Client Service or Datacap Rulerunner Quattro.

Overview of Taskmaster Email Actions Installation

When installing in a Workstation/Mail Server configuration, you install the Taskmaster Email Actions only on the Workstation.

When installing in a Client/TMServer/Mail Server configuration, you install the Taskmaster Email Actions on the TMServer hosting the application (on which the application's RRX files are found - usually found in C:\Datacap\RRS\RRX) before you can add the actions to the application. You also install the actions on all Workstations or Taskmaster Servers that run these Email Actions using Taskmaster Client, Taskmaster Client Service, Datacap Studio, or Datacap Rulerunner Quattro (when using Quattro, tasks containing email actions should only be set up to run in a single thread).

The installation process is made up of the following steps:

1. Ensure you have administrator access to all machines, and ensure you have a working Taskmaster 7.5 SP1 or later client/server environment.
2. Make a note of and shut down all Taskmaster software on all machines in your environment.
3. Make the Taskmaster Email installation package available on your network .
4. Install the Taskmaster Email Actions on all Rulerunner Workstations that are running email tasks, on Datacap Studio Workstations, and on the TMServer hosting the application that is to use Taskmaster Email.
5. Using Datacap Studio, incorporate Taskmaster Email actions in your application.
6. Restart Taskmaster software on all machines.

Make a Note of and Shut Down Taskmaster Software

Before beginning the installation of the Taskmaster Email Actions on any machine in your network, we recommend identifying which Taskmaster software is running on each machine

(so you will know what to start back up after installing the software), and then shutting down all Taskmaster software on all machines in the following sequence:

1 Taskmaster Client software for all applications (including Taskmaster Client, Taskmaster Web Client, Taskmaster Client Service, Datacap Studio, Datacap Rulerunner Quattro, Batch Pilot, Report Viewer, etc.)

2 Taskmaster Web and any Taskmaster Web Services (WRRS, Fingerprint Service, etc.)

3 Taskmaster Server Service

Install Taskmaster Email Actions

This procedure provides instructions on how to run the Datacap Taskmaster Email Actions InstallShield Wizard. Install the Taskmaster Email Actions on all Workstations or Taskmaster Servers that run these Email Actions using Taskmaster Client, Taskmaster Client Service, or Datacap Rulerunner Quattro, or where Datacap Studio is used.

WARNING

While the installation package (Setup.exe) can be located anywhere (for example, on an accessible network drive, on the CD in the Workstation's CD/DVD drive), the destination of the installed software must be on a physical hard drive of the machine from which you run the Setup.exe.

1 Make the installation package available on your network, or insert the Taskmaster Email Actions CD in the Workstation’s CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe.
When UAC is on, the User Account Control window opens. Click Allow.
The Datacap Taskmaster Email Actions Setup InstallShield Wizard opens.

2 The InstallShield Wizard displays a Preparing to Install... window followed by the InstallShield Wizard Welcome window.

3 Click Next. The License Agreement window opens.

4 Click I accept the terms in the license agreement and click Next. The Ready to Install the Program window is displayed.

5 Click Install. The Installation Wizard displays progress messages and a progress indicator, and when complete, displays the InstallShield Wizard Complete window.

6 Click Finish. The InstallShield Wizard closes.

Email Actions and Taskmaster Smart Parameters

Several of the Email Actions accept parameters that can be either static arguments or variables called Taskmaster Smart Parameters. During processing Smart Parameters are replaced with specific values from the item being processed, for example, when sending an email, the name and path of a file to attach to the email. In this Guide, the explanations of the actions that accept Smart Parameters explicitly state "Smart Parameters are supported." This information is also included in the action's help that you can access while working in Datacap Studio.

For the most up to date documentation on Smart Parameters, see:

http://dnld.datacap.com/TechNotes/SmartParameters.htm

Incorporate Taskmaster Email Actions in Application

Once you have successfully installed the Taskmaster Email Actions on the TMServer, the actions are available for you to incorporate into your Datacap Studio applications.

Instructions for Adding Taskmaster Email Actions to Application

In general, the high-level steps to incorporate Taskmaster Email Actions in an application are as follows:

1. Skip this step if installing in a Workstation/Mail Server configuration.
   On the Workstation, ensure the Workstation's HKEY_LOCAL_MACHINE > SOFTWARE > DATACAP > RRS> locations > rrs registry entry reflects the UNC name of the TMServer (rather than C:\) and the path to the \\TMServer\Datacap\RRS folder where the Taskmaster Email Actions are installed.
2. Start Taskmaster Server Service
   - In Workstation/Mail Server configuration, start Taskmaster Server Service on Workstation.
   - In Client/TMServer/Mail Server configuration, start Taskmaster Server Service on TMServer.
3. On the Workstation, start Datacap Studio and open the application.
4. When Datacap Studio opens, on the Rulemanager > Actions Library tab, scroll down, and ensure Email, IMail and EWSMail are listed under Global Actions.
5. On the Rulemanager > Rulesets tab, select the ruleset to which you want to add a new email rule or email action, then click Lock ruleset for editing. The lock icon changes, and a red checkmark appears next to the ruleset name.
6. Click the Sync DCO view with Ruleset view button to expand the Document Hierarchy and highlight the object(s) to which the ruleset is bound. Note the level of the object to which the ruleset is bound - see the Datacap Studio action-level help or the Actions Reference sections below for information on which actions work at which levels, and on what each action does.
7. Incorporate the appropriate actions in the ruleset.
8. Click Save to save the changes to the ruleset.
9. Test your changes, and when ready, on the Rulemanager > Rulesets tab, click Publish ruleset.

Restart Taskmaster Software

After incorporating Taskmaster Email Actions in and testing your application, restart the Taskmaster software that was previously running on other machines. Restart Taskmaster in the following sequence:

1. Taskmaster Server Service
2. Taskmaster Web and any Taskmaster Web Services (WRRS, Fingerprint Service, etc.)
3. Taskmaster Client software (including Taskmaster Client, Taskmaster Web Client, Taskmaster Client Service, Datacap Rulerunner Quattro Service, Datacap Studio, Batch Pilot, Report Viewer, etc.)

IMail.RRX Input Actions Reference

Use the actions in IMail.RRX when your mail server uses the Internet Message Access Protocol (IMAP); these include Microsoft Exchange Server, Novell GroupWise, and others.

Each email message (containing one or more of the desired attachment types) becomes a new Taskmaster document with the message headers and body, and pages are created for each of the attachments.

IMail sets an EmailCount at the Batch level, and sets the following Document level variables for each email message accepted (Done): TYPE="Document", Subject, From, To, DateSent, Priority, Body=<email body>, User.

IMail sets the following Page level variables for each attachment: TYPE="Other", IMAGEFILE=<attachment filename>.

im_login

Specifies the IMAP mail server URL and email account login credentials.

Actions Library	IMail.RRX
Description	Specifies the mail server URL and login credentials. Call once for each batch. Connects to the mail server using the specified account information. Login credentials are for a mail server that supports the IMAP protocol (such as MS Exchange, and many others). Refer to the top-level help for general information about scanning the inbox and requirements for the "Done" and "Problem" folders.
Parameters	hostname - URL of IMAP mail server username - Username for mail account password - Password for mail account (SIT encryption is supported)
Level	Batch level Open event only.
Returns	**True** if the login succeeds. Otherwise, **False**.
Example	`im_login("mymailserver.com","theuser","password")`

im_scan

Contact the specified mail server for incoming email with image attachments.

Actions Library	IMail.RRX
Description	Scans emails in the Inbox for specified attachments, imports selected emails with attachments into the batch. Call once for each batch. Each input email will contain the following variables set in the document hierarchy: **TYPE**: Always set to "Document". **Subject**: Email subject. **Body**: Text within the email. **DateSent**: Sent date stamp on the email. **From**: Email sender. **User**: Email user. **Priority**: State of the email importance flag. **To**: Email recipients. Each email attachment will have the following variables set: **TYPE**: Always set to "Other". **IMAGEFILE**: Name of the attachment.
Parameters	None
Level	Batch level Open event only.
Returns	Returns **False** if the operation fails, and pauses before returning. Otherwise, **True**. If no selected emails were available, the action returns **True** and pauses before returning. Action returns when timeout is reached, or the requested number of emails have been processed.
Example	`im_scan()`

im_logout

Disconnect from the mail server.

Actions Library	IMail.RRX
Description	Disconnects from the mail server. Call once for each batch after im_login and im_scan have completed.
Parameters	None.
Level	Batch level Open or Close event only.
Returns	Always **True**.
Example	`im_logout()`

im_types

Specifies valid file extensions for image attachments.

Actions Library	IMail.RRX
Description	This action specifies the allowable email attachment types. If this action is not called, the default value of ".pdf" will be used. If this action is called and no attachment types are specified, all emails and any attachments are added to the batch. When emails without attachments are processed, this will result in documents without any "pages." If attachment types are specified and the email contains: • Only the specified attachment types, the email and attachments will be added to the batch • Only unspecified attachment types, or if it contains a mix of specified and unspecified attachment types, the email is moved to the Problem folder • No attachments, the email is moved to the Problem folder
Parameters	Comma-separated list of image file extensions to import, with or without period. Can be empty (all emails with or without attachments will be processed).
Level	Batch level.
Returns	Always **True**.
Example	`im_types("jpg,tif")`

im_wait_time

Specifies the maximum amount of time, in seconds, to wait for input emails for a single batch.

Actions Library	IMail.RRX
Description	Maximum amount time to wait for input emails for a single batch. Used by the im_scan action after the first email has been processed to determine how long to wait for the batch to fill up. The import of emails into the batch will stop when the wait limit is reached or when the maximum number of emails per batch has been reached. While waiting for new mail to arrive, the mailbox will be polled every two seconds to check for waiting mail. The action will continue to include new emails into the batch until this wait time is reached or the maximum number of emails per batch is reached. If this action is not called, the default wait time of 5 seconds will be used.
Parameters	Maximum number of seconds to wait.

Level	Batch level.
Returns	Always **True**.
Example	`im_wait_time("20")`

im_abort_time

Specifies time delay, in seconds, before returning when a batch aborts.

Actions Library	IMail.RRX
Description	Amount of time to wait before returning when an abort occurs. This action can be useful to prevent a large number of aborted batches due to an abort condition. For example, if the email server should become unavailable for a time, the abort timeout will limit the number of aborted batches until the mail server becomes available again. If this action is not called, the default time value of 30 seconds will be used.
Parameters	Number of seconds to wait before returning after an abort.
Level	Batch level.
Returns	Always **True**.
Example	`im_abort_time("60")`

im_max_docs

Specifies maximum number of emails to include in a single batch.

Actions Library	IMail.RRX
Description	The import of emails into the batch will stop when this limit is reached or when the maximum wait time has been reached. While waiting for new mail to arrive, the mailbox will be polled every two seconds to check for waiting mail. If this action is not called, the default value of 100 will be used. The actual number of emails included in the batch could be less than this maximum.
Parameters	Maximum number of emails in each batch.
Level	Batch level.
Returns	Always **True**.
Example	`im_max_docs("200")`

im_done_folder

Specifies destination IMAP folder for successfully imported emails.

Actions Library	IMail.RRX
Description	When an email is processed and the attachment is successfully imported, the email is moved to the folder name specified by this action. If this action is not called, the default value of "Done" will be used. The folder is expected to be at the root level, the same level as

	the inbox folder in the mail account specified by the im_login action. If the folder exists in a subfolder, then use a forward slash to separate the folder names.
Parameters	Folder name
Level	Batch level.
Returns	Always **True**.

Example	This example changes the default name of the "Done" folder to "Imported" - and requires the "Imported" folder to be on the same level as the "Inbox" folder:
	`im_done_folder("Imported")`
	This example specifies a folder which is a subfolder of the Inbox folder:
	`im_done_folder("Inbox/Imported")`

im_problem_folder

Specifies destination IMAP folder for problem emails.

Actions Library	IMail.RRX
Description	When an email is processed and the attachment is not one of the expected types, the email is moved to the folder specified by this action. If this action is not called, the default value of "Problem" will be used. The folder is expected to be on the root level, the same level as the inbox folder of the mail account specified by the im_login action. If the folder exists in a subfolder, then use a forward slash to separate the folder names.
Parameters	Folder name
Level	Batch level.
Returns	Always **True**.

Example	This example changes the default name of the "Problem" folder to "Failed" - and requires the "Failed" folder to be on the same level as the "Inbox" folder:
	`im_problem_folder("Failed")`
	This example specifies the folder which is a subfolder of the Inbox folder:
	`im_problem_folder("Inbox/Failed")`

IMail Input Actions Sample Output

This example rrsvscan.xml file shows the results of successfully processing an incoming email that contained two email attachments with the IMail Input Actions.

```
<?xml-stylesheet type="text/xsl" href="..\..\dco.xsl"?>
<B id="20100141.005">
      <V n="TYPE">APT</V>
      <V n="EmailCount">5</V>
```

```
      <D id="20100141.005.1">
            <V n="TYPE">Document</V>
            <V n="Subject">tif,xls</V>
            <V n="DateSent">Fri, 21 May 2010 07:49:09 -0400</V>
            <V n="From">Peggy Shea&lt;pshea@abc.com&gt;</V>
            <V n="Priority">3</V>
            <V n="To">&lt;demo_user@defg.com&gt;</V>
            <V n="Body">_____
From: Peggy Shea [mailto:pshea@abc.com]
Sent: Thursday, May 20, 2010 2:18 PM
To: &apos;qatesting@defg.com&apos;
Subject: tif,xls</V>
            <V n="User">demo_user</V>
            <V n="STATUS">0</V>
            <P id="P1.1">
                  <V n="TYPE">Other</V>
                  <V n="IMAGEFILE">tm000001.tif</V>
                  <V n="STATUS">0</V>
            </P>
            <P id="P1.2">
                  <V n="TYPE">Other</V>
                  <V n="IMAGEFILE">tm000002.xsl</V>
                  <V n="STATUS">0</V>
            </P>
      </D>
</B>
```

EWSMail.RRX Input Actions Reference

Use the EWSMail Input Actions when your mail server is an MS Exchange 2007 or above mail server, configured to allow Exchange Web Service access.

Each email message (optionally containing one or more desired attachment types) becomes a new Taskmaster document with the message headers and body, and pages are created for each of the attachments.

EWSMail sets an EmailCount at the Batch level, and sets the following Document level variables for each email message accepted (Done): TYPE="Document", MessageID, Subject, From, To, DateSent, DateReceived, Priority, Body.

When you don't use the ex_EMLOption action, EWSMail sets the following Page level variables for each attachment: TYPE="Other", IMAGEFILE=<attachment filename in batch>, ATTACHNAME=<original attachment filename>.

When you use the ex_EMLOption action, pages for attachments are not created, and variables for those attachments are not set.

ex_ews_version

Select the Exchange Server version.

Actions Library	EWSMail.RRX
Description	Each version of Exchange uses a slightly different communication protocol. Use this action to set the expected version. In order to connect successfully with: • Exchange 2007 SP1, call this action with parameter "1" before im_login. • Exchange 2010, call this action with parameter "2" before im_login. • When called with any other parameter, it means "the latest version known by the .NET library in use" (which is .NET 3.5 and is currently Exchange 2010). If this action is not called at all, defaults to the latest version.
Parameters	1=Exchange 2007 SP1 2=Exchange 2010 0=latest version (default)
Level	Batch level, Open event.
Returns	Always returns **True.**
Example	`ex_ews_version("1")` `ex_login("mymailserver/Exchange.asmx","Username@Org","password")`

ex_login

Attach to the Exchange Server and login to an email account.

Actions Library	EWSMail.RRX
Description	Connects to the mail server using the specified account information. Login credentials are for a Microsoft Exchange mail server. The mail account must contain an Inbox folder and separate folders for messages imported (Done) and errors (Problem). Done and Problem folders must be subfolders of the email account's Inbox. The names of these folders can be specified using the ex_done_folder and ex_problem_folder actions. The Microsoft Exchange Email actions are designed to scan an email Inbox for incoming mail messages, and place selected messages into a new batch. It is possible to ignore all messages except those containing specific attachment types. The actions are typically assigned to a Task that is executed by an unattended Taskmaster Rulerunner station. Multiple Inboxes can be scanned by stringing together a set of login/scan/logout actions for each Inbox, or by assigning an Inbox to each input task
Parameters	string **hostname** - URL of Exchange Web Service, ends with /Exchange.asmx string **username** - Username@Org for mail account and organization (blank to use Windows Authentication) string **password** - Password for mail account (blank if none or Windows Authentication, SIT encryption is supported)
Level	Batch level, open event.
Returns	**True** if the login succeeds. Otherwise, **False**.
Example	`ex_login("mymailserver/Exchange.asmx","Username@Org","password")`

ex_scan

Contact the specified mail server for incoming email with image attachments.

Actions Library	EWSMail.RRX
Description	Scans emails in the Inbox for specified attachments, imports selected emails with attachments into the batch. A connection to the email server must have previously been established using the ex_login action. Each captured email (document) will contain the following variables set in the document hierarchy: **TYPE** : Always set to "Document". **MessageID** : Email message ID.

	DateSent: Sent date stamp on the email. **DateReceived**: Received date stamp on the email. **Subject** : Email subject. **Body** : Text within the email. **From** : Email sender. **Priority** : State of the email importance flag. **To** : Email recipients. Each email attachment (page) will have the following variables set: **TYPE** : Always set to "Other". **IMAGEFILE** : Name of the attachment as saved on disk. **ATTACHNAME** : Name of the attachment
Parameters	None
Level	Batch level Open event only.
Returns	Returns **False** if the operation fails, and the action will also pause before returning based on the configured abort time configured by ex_abort_time. Otherwise, **True.** If no selected emails were available, the action returns **True** and also pauses before returning based on the wait time configured using ex_wait_time.
Example	`ex_scan()`

ex_logout

Disconnect from the mail server.

Actions Library	EWSMail.RRX
Description	Closes the connection to the mail server. Call once for each batch after ex_login and ex_scan have completed.
Parameters	None.
Level	Batch level Open or Close event only.
Returns	Always **True**.
Example	`ex_logout()`

ex_types

Specifies valid image attachment file extensions.

Actions Library	EWSMail.RRX
Description	This action specifies the allowable email attachment types. If this action is not called, the default value of ".pdf" will be used. If this action is called and no attachment types are specified, all emails and any attachments are added to the batch. If attachment types are specified and the email contains:

	• Only the specified attachment types, the email and attachments will be added to the batch • Only unspecified attachment types, or if it contains a mix of specified and unspecified attachment types, the email is moved to the Problem folder • No attachments, the email is moved to the Problem folder
Parameters	Comma-separated list of file extensions to import. Extensions can be specified with or without the period. If called with an empty parameter (""), ex_scan will import emails even when they have no attachments.
Level	Batch level.
Returns	Always **True**.
Example	`ex_types("jpg,tif")`

ex_wait_time

Specifies the maximum amount of time, in seconds, to wait for input emails for a single batch.

Actions Library	EWSMail.RRX
Description	Maximum time to wait for input emails for a single batch. Used by the ex_scan action after the first email has been processed to determine how long to wait for the batch to fill up. The import of emails into the batch will stop when the wait limit is reached or when the maximum emails per batch has been reached. While waiting for new mail to arrive, the mailbox will be polled every two seconds to check for waiting mail. The action will continue to include new emails into the batch until this wait time is reached or the maximum number of emails per batch is reached. If this action is not called, the default wait time of 5 seconds will be used.
Parameters	Maximum number of seconds to wait.
Level	Batch level Open event only.
Returns	Always **True**.
Example	`ex_wait_time("20")`

ex_abort_time

Specifies time delay, in seconds, before returning when a batch aborts.

Actions Library	EWSMail.RRX
Description	Amount of time to wait before returning when an abort occurs. This action can be useful to prevent a large number of aborted batches due to an abort condition. For example, if the mail server should become unavailable for a time, the abort timeout will limit the amount of aborted batches until the mail server becomes available again. If this action is not called, the default value of 30 seconds will be used.
Parameters	Number of seconds to wait

Level	Batch level.
Returns	Always **True**.
Example	`ex_abort_time("60")`

ex_max_docs

Specifies the maximum number of emails to include in a single batch.

Actions Library	EWSMail.RRX
Description	The import of emails into the batch will stop when this limit is reached or when the maximum wait time has been reached. While waiting for new mail to arrive, the mailbox will be polled every two seconds to check for waiting mail. If this action is not called, the default value of 100 will be used. The actual number of emails included in the batch could be less than this maximum.
Parameters	Maximum number of emails in each batch.
Level	Batch level Open event only.
Returns	Always **True**.
Example	`ex_max_docs("50")` `ex_scan()`

ex_done_folder

Specifies the destination folder for successfully imported emails.

Actions Library	EWSMail.RRX
Description	When an email is processed and the attachment is successfully imported, the email is moved to the folder name specified by this action. If this action is not called, the default value of "Done" will be used. This folder must be a subfolder of the email account's Inbox. If this folder does not exist, items are sent to the Deleted folder after processing.
Parameters	Folder name
Level	Batch level.
Returns	Always **True**.
Example	`ex_done_folder("Imported")`

ex_problem_folder

Specifies the destination folder for problem emails.

Actions Library	EWSMail.RRX
Description	When an email is processed and the attachment is not one of the expected types, the email is moved to the folder specified by this action. If this action is not called, the default value of "Problem" will be used.

	This folder must be a subfolder of the email account's Inbox. If this folder does not exist, items are sent to the Deleted folder after processing.
Parameters	Folder name
Level	Batch level
Returns	Always **True**.
Example	`ex_problem_folder("Failed")`

ex_EMLOption

Creates a one page document per email containing an .eml file.

Actions Library	EWSMail.RRX
Description	If set, the ex_scan action creates a one page document containing the email and attachments in an .eml file; no attachment pages are created. When called with a non-zero parameter, the ex_scan function does not create pages for each attachment, instead one page is created per email document, containing an .eml file suitable for subsequent processing using eDocument Conversion actions. When you use this action, pages for attachments are not created, and variables for those attachments are not set.
Parameters	Optional - use a nonzero value to store one .eml file per email
Level	Batch level
Returns	Always **True**.
Example	`ex_EMLOption(1)`

EWSMail Input Actions Sample Output

This example rrsvscan.xml file shows the results of successfully processing an incoming email that contained two email attachments with the EWSMail Input Actions.

```
<?xml-stylesheet type="text/xsl" href="..\..\dco.xsl"?>
<B id="20100138.012">
      <V n="TYPE">APT</V>
      <D id="20100138.012.1">
            <V n="TYPE">Document</V>
            <V n="MessageID">&lt;61A4CE6BE429F4488D511A47@mailbox.ABC.COM&gt;</V>
            <V n="Subject">PDFs you requested</V>
            <V n="Priority">1</V>
            <V n="From">Sales Department</V>
            <V n="To">Contracts Department</V>
            <V n="STATUS">0</V>
            <P id="P1.1">
                  <V n="TYPE">Other</V>
```

```
                <V n="IMAGEFILE">tm000001.pdf</V>
                <V n="ATTACHNAME">ResellerLicense.pdf</V>
                <V n="STATUS">0</V>
            </P>
            <P id="P1.2">
                <V n="TYPE">Other</V>
                <V n="IMAGEFILE">tm000002.pdf</V>
                <V n="ATTACHNAME">DirectLicense.pdf</V>
                <V n="STATUS">0</V>
            </P>
        </D>
</B>
```

Email.RRX Sending Actions Reference

Use the Email.RRX Sending Actions to construct and send emails using an SMTP mail server.

SendEMail

Sends an email.

Actions Library	**Email.RRX**
Description	Sends an email assembled by previous actions. Typically, this is the final action in an **EMail** ruleset. At a minimum, the SetSender and SetRecipients actions must be called prior to sending an email. After sending, this action will discard the contents of the email in memory. Calls to the email actions after SendEMail will cause the creation of a new email message.
Parameters	None
Level	All levels.
Returns	**False** if the rule does not include a previous **SetRecipients** action, or if the email cannot be sent. Otherwise, **True**. If the email cannot be sent, the batch will be set to abort.

Example	`SetSender("paul@adomain.com")`
	`SetRecipients("lisa@adomain.com,beth@adomain.com")`
	`SetSubject("Document Integrity")`
	`SetEMailBody("Document Page Types and counts are accurate.`
	`Thanks for your help.")`
	`SendEMail()`

SetAttachment

Adds a file attachment to the email.

Actions Library	**Email.RRX**
Description	Identifies the name and path of a file to attach to the current email.
Parameters	The file's path and name. Smart Parameters are supported.
Level	All levels.
Returns	**False** if the file does not exist or cannot be attached. Otherwise, **True**.
Example	This example attaches the Export file of the current batch to the email. `SetAttachment("h:\APT\export\+@BATCHID+.txt")`

SetBlindCarbonCopyRcpts

Sets the Blind Carbon Copy recipients' email address(es).

Actions Library	**Email.RRX**
Description	Adds email addresses to the **Bcc** (Blind Carbon Copy) portion of the email's header.

Parameters	Email addresses to receive a copy of the email as a blind carbon copy. You can enter multiple email addresses separated by commas.
Level	All levels.
Returns	**False** if you do not enter an email addresses parameter, if the address is rejected by the mail system, or if the email object cannot be initialized. Otherwise, **True**. Invalid email addresses may not be reported until SendEMail is called.
Example	`SetBlindCarbonCopyRcpts(james@adomain.com)`
See also	SetCarbonCopyRcpts, SetRecipients

SetCarbonCopyRcpts

Sets the Carbon Copy recipients' email address(es).

Actions Library	**Email.RRX**
Description	Adds email addresses to the **Cc** (Carbon Copy) portion of the email's header.
Parameters	Email addresses to receive a copy of the email as a carbon copy. You can enter multiple email addresses separated by commas.
Level	All levels.
Returns	**False** if you do not enter an email addresses parameter, if the address is rejected by the mail system or if the email object cannot be initialized. Otherwise, **True**. Invalid email addresses may not be reported until SendEMail is called.
Example	`SetRecipients("lisa@adomain.com")`
See also	SetBlindCarbonCopyRcpts, SetRecipients

SetEMailBody

Sets the text of the email's body.

Actions Library	**Email.RRX**
Description	Sets the text of the email's body.
Parameters	The email message text. Smart Parameters are supported.
Level	All levels.
Returns	**False** if the mail object cannot be initialized. Otherwise, **True**.

Example	`SetSubject("Document Integrity")`
	`SetEMailBody("Document Page Types and counts are accurate.`
	`Batch ID was @BATCHID. Thanks for your help.")`

See also	SetAttachment

SetMailServer

Sets the mail server to use for sending mail.

Actions Library	**Email.RRX**
Description	Sets the address of the outgoing mail (SMTP) server. This may be the same mail server you configure in your mail program. The server must be accessible from the computer running the email actions. This action should be the first action in an email rule if the CDOSYS object is being used. Use this action only if you are sending emails with CDOSYS. To use CDOSYS, this action must be called prior to any of the other email actions. If this action is not called prior to other email actions, these actions will use Outlook for sending emails. You can use Email actions to direct a task to compose and send emails containing information and attachments. Email actions use the Windows CDOSYS library to send email via your preferred SMTP mail server. The CDOSYS object is included with Windows 2000 and above. Alternatively, Email actions can use the Outlook object but this is not recommended. One of these two libraries (CDOSYS or Outlook) must be registered on the computer that runs rules employing email actions. Outlook is primarily useful for demonstration purposes, as it is not suitable for unattended operation: it requires the Outlook user to be logged into the computer, and security prompts may be displayed for each message sent.
Parameters	The IP or DNS address of the outgoing mail (SMTP) server.
Level	All levels.
Returns	Always **True**
Example	`SetMailServer("mail.YourISP.com")`

SetRecipients

Sets the email recipients' address(es).

Actions Library	**Email.RRX**
Description	The email address of the email's primary recipient(s). You can either call this action multiple times to add multiple recipients, or you can enter multiple email addresses separated by commas.
Parameters	Email address(es) of recipient(s). To enter multiple email addresses, separate addresses with commas.
Level	All levels.

Returns	**False** if you do not enter an email addresses parameter, if the address is rejected by the mail system, or if the email object cannot be initialized. Otherwise, **True**. Invalid email addresses may not be reported until SendEMail is called.
Example	`SetRecipients("lisa@adomain.com,joe@adomain.com")`
See also	SetCarbonCopyRcpts, SetBlindCarbonCopyRcpts

SetSender

Sets the sender's email address.

Actions Library	**Email.RRX**
Description	Sets the email address of the sender for the current email. When using the **CDOSYS** object, use this action. When using the **Outlook** object, the current email account is used as the sender.
Parameters	The sender's email address.
Level	All levels.
Returns	**False** if the mail object cannot be initialized. Otherwise, **True**. Invalid email addresses may not be reported until SendEMail is called.
Example	`SetSender("paul@adomain.com")`
See also	SetRecipients

SetSubject

Sets the text for the email's Subject field.

Actions Library	**Email.RRX**
Description	Sets the text for the email Subject field. Smart Parameters are supported.
Parameters	The subject line of the email. Smart Parameters are supported.
Level	All levels.
Returns	**False** if the mail object cannot be initialized. Otherwise, **True**.
Example	`SetSubject("Document Integrity")`

Uninstall Taskmaster Email Actions

This section provides the information you need to uninstall Taskmaster Email Actions. This involves completing processing of all batches, shutting down Taskmaster client and web software, using Datacap Studio to remove all Taskmaster Email Actions from all applications, shutting down all Taskmaster services, using Windows Add or Remove Programs or the Taskmaster Email Actions InstallShield Setup.exe to remove the Taskmaster Email files from all machines, and restarting Taskmaster.

Ensure All Batches Have Been Completed

Before you begin uninstalling, use the Taskmaster Client Job Monitor to ensure that all batches have been completely processed.

Ensure All Taskmaster Clients are Shut Down

Before you begin uninstalling, ensure that all Taskmaster Client and Taskmaster Web Client users on all workstations are logged off.

Ensure Taskmaster Web is Shut Down

Before you begin uninstalling, ensure that Taskmaster Web and any Taskmaster Web Services (WRRS, Fingerprint Service, etc.) have been shut down.

Remove Taskmaster Email Actions from Applications

Using Datacap Studio, remove all Taskmaster Email Actions from all applications.

Stop Taskmaster Client Service

This procedure provides instructions on how to stop the Taskmaster Client Service.

1. From your Windows Start menu, select Programs, select Datacap , select Taskmaster Client, and click Taskmaster Client Service. The TMClient Service settings window opens.
2. When the Status:
   - Shows as Stopped, continue with the next step.
   - Shows as Running, click Stop. A Stopping Service message is displayed. The Status changes to Stopped when the service is stopped.
3. Close the Services window.

Stop Datacap Rulerunner Quattro Service

This procedure provides instructions on how to stop the Datacap Rulerunner Quattro Service. If you attempt to stop the Quattro Service while it is processing one or more job tasks, the service will complete its current work before shutting down.

1. From the Server's Windows Start menu, select Settings, select Control Panel, select Administrative Tools, select Services.

2 Select Datacap Rulerunner Quattro and select Stop. A Service control message box is displayed, then closes when the service is stopped.

3 Close the Services window.

Stop or Ensure Taskmaster Server Service is Stopped

This procedure provides instructions on how to stop or ensure the Datacap Taskmaster Server Service is stopped on the Server.

1 From the Server's Windows Start menu, select Programs, select Datacap, select Taskmaster Server, and click Taskmaster Server. The TMS Configuration window opens.

2 When the Status:

- Shows as Stopped, continue with the next step.
- Shows as Running, click Stop. A Stopping Service message is displayed. The Status changes to Stopped when the service is stopped.

3 Close the TMS Configuration window.

Uninstall Taskmaster Email Actions

1 You can uninstall Taskmaster Email Actions either by using Windows Add or Remove Programs (remove the Datacap Taskmaster Email Actions 7.6 entry), or by running the Taskmaster Email Actions Setup.exe.

2 To remove the actions using the Setup.exe, make the installation package available on your network, or insert the Taskmaster Email Actions CD in the Workstation's CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe.
When UAC is on, the User Account Control window opens. Click Allow.
The Datacap Taskmaster Email Actions Setup InstallShield Wizard opens.

3 The InstallShield Wizard displays a Preparing to Install... window followed by the InstallShield Wizard Welcome window.

4 Click Next. The Program Maintenance window opens with the Modify option selected.

5 Select the Remove option and click Next. The Remove the Program window opens.

6 Click Remove. The Wizard displays progress messages and a progress indicator, and when complete, displays the InstallShield Wizard Complete window.

7 Click Finish. The InstallShield Wizard closes.

Restart Taskmaster Software

After uninstalling Taskmaster Email Actions, restart the Taskmaster software that was previously running on other machines. Restart Taskmaster in the following sequence:

1 Taskmaster Server Service

2 Taskmaster Web and any Taskmaster Web Services (WRRS, Fingerprint Service, etc.)

3 Taskmaster Client software (including Taskmaster Client, Taskmaster Web Client, Taskmaster Client Service, Datacap Rulerunner Quattro Service, Datacap Studio, Batch Pilot, Report Viewer, etc.)

Actions Index