

*ariadne*

where *i* is for information



*CoolSpools*

EMAIL

## *User Guide*

*Version V6R1M0  
2010*

*ariadne software ltd, cheltenham, england*

# ***Table of Contents***

Introduction .....	10
Why Choose CoolSpools Email? .....	11
Prerequisites .....	12
Upgrade Notes .....	13
License Keys .....	13
Warning/Disclaimer.....	14
Minimum OS/400 Release Level .....	14
Product Library .....	15
Memo to Users .....	15
What's new in this release?.....	16
Important Note about POP settings!.....	18
Address Lists.....	19
Address Directories .....	20
Installation .....	21
Maintenance.....	22
Setup.....	23
How CoolSpools Email works .....	23
1. Mail Server Framework (MSF) .....	23
2. CoolSpools SMTP.....	24
Telling CoolSpools Email which method to use .....	24
Configuring CoolSpools Email to use Mail Server Framework .....	25
Configuring CoolSpools Email to use CoolSpools SMTP .....	26
CFGSMTPSVR (Configure SMTP Server) command.....	26
Autostart servers (AUTOSTART).....	27
Number of initial servers (NBRSVR) .....	27
Run as user profile (USER).....	28

Data queue library (DTAQLIB) .....	28
Write log records (LOG) .....	28
Report errors to message queue (ERRMSGQ).....	29
Report errors to email address (ERREMAIL) .....	29
Report non-delivery to msgq (DSNMSGQ) .....	30
Report non-delivery to email (DSNEMAIL).....	30
Autostart in subsystem (SBSD).....	31
Class (CLS).....	32
Submit to job queue (JOBQ) .....	33
First-level retries (LVL1RETRY).....	33
Second-level retries (LVL2RETRY).....	34
Third-level retries (LVL3RETRY).....	35
Host name (HOSTNAME) .....	35
Domain name (DMNNAME) .....	35
Mail server name or IP address (MAILSERVER) .....	36
Server authentication (SVRAUTH).....	36
Server login (SVRLOGIN) .....	38
Managing the CoolSpools SMTP servers .....	39
STRSMTPSVR (Start SMTP Server) command .....	39
Number of server jobs to start (NBRJOBS).....	39
Start retry server if inactive (RTYSVR).....	40
ENDSMTPSVR (End SMTP Server) command .....	40
WRKSMTPMSG (Work with SMTP Messages) command .....	40
Option (OPTION) .....	41
Using CoolSpools Email.....	43
The CoolSpools Email Menu .....	43
SNDCMNMSG command .....	44

EMAILID - Email id to retrieve .....	44
TO - Recipient(s).....	44
FROM - Sender.....	46
REPLYTO - Send replies to .....	49
SUBJECT - Subject.....	49
MSG – Message text.....	49
ATTACH - Attachments.....	51
MSGSTMF - Stream file containing message .....	54
PRIORITY - Priority.....	56
METHOD - Method .....	56
CFMDEL - Confirm delivery .....	57
SNDMLTMSG - Send multiple messages? .....	57
SAVE - Save options.....	58
LOG - Logging options .....	59
PWD - Encryption password .....	59
CHARSET – Character set encoding.....	60
OPTIONS - Miscellaneous options.....	61
CRTCMNMSG command .....	63
RSNCMNMSG command.....	67
EMAILID.....	67
DLTCMNMSG command.....	68
DLTCMNMSG command.....	68
DLTSAVMSG .....	68
DLTLOGDTA.....	68
Working with Email Address Lists.....	69
Creating an address list.....	72
ADRL .....	72

TEXT .....	72
Example: .....	72
Changing an address list .....	73
ADRL .....	73
TEXT .....	73
Example: .....	73
Copying an address list .....	74
FROMADRL .....	74
TOADRL .....	74
TEXT .....	74
CPYADRLE .....	74
Deleting an address list .....	75
ADRL .....	75
Converting a Distribution List to an Address List .....	76
DSTL .....	76
ADRL .....	76
TEXT .....	76
HOSTNAME .....	76
DOMAIN .....	77
Working with Address List Entries .....	79
ADRL .....	79
Adding an address list entry .....	81
ADRL .....	81
EMAIL .....	81
NAME .....	81
TYPE .....	81
Changing an address list entry .....	83

ADRL .....	83
EMAIL .....	83
NAME.....	83
TYPE.....	83
Copying address list entries.....	84
FROMADRL.....	84
TOADRL .....	84
Removing an address list entry .....	84
ADRL .....	84
EMAIL .....	84
Working with Email Address Directories .....	86
Creating an address directory.....	88
ADRDIR .....	88
TEXT .....	88
Changing an address directory .....	89
ADRDIR .....	89
TEXT .....	89
Copying an address directory .....	90
FROMADRDIR.....	90
TOADRDIR .....	90
TEXT .....	90
CPYADRDIRE.....	90
Deleting an address directory .....	91
ADRDIR .....	91
Working with Address Directory Entries.....	92
ADRDIR .....	92
Adding an address directory entry .....	94

ADRDIR .....	94
EMAIL .....	94
FIRSTNAME .....	94
LASTNAME.....	94
PREFIX .....	94
MIDDLENAME .....	94
SUFFIX .....	94
COMPANY .....	95
POSITION.....	95
EMAILNAME .....	95
Changing an address directory entry .....	96
ADRDIR .....	96
EMAIL .....	96
Copying address directory entries .....	97
FROMADRDIR.....	97
TOADRDIR .....	97
ADRDIR .....	98
EMAIL .....	98
SNDCMNICAL command .....	99
SUMMARY .....	99
STATUS.....	99
START .....	99
DURATION .....	99
LOCATION.....	100
ORGANIZER.....	100
DESCRIPT .....	100
DSCSTMF .....	100

ATTENDEES.....	101
ATTACH.....	102
PRIORITY .....	105
CLASS .....	105
CATEGORIES.....	105
SNDCMNVCAL command.....	106
ATTENDEES.....	106
IMPEMLDTA command .....	108
Restrictions .....	108
FROMLIB .....	108
REPLACE .....	108
Using the CoolSpools Email Send API CM_SNDAPIR.....	110
Required Parameter Group.....	113
Optional Parameter Group 1 .....	118
Optional Parameter Group 2 .....	118
Optional Parameter Group 3 .....	118
Optional Parameter Group 4 .....	118
Optional Parameter Group 5 .....	119
Optional Parameter Group 6 .....	120
Optional Parameter Group 7 .....	120
Optional Parameter Group 8 .....	121
Optional Parameter Group 9 .....	122
Required Parameter Group.....	125
The CoolSpools Email Database .....	126
Message Log CM_MSGLOG .....	126
Message Recipients CM_MSGRCP.....	126
Message Attachments CM_MSGATC.....	127

Environment variables .....	128
-----------------------------	-----

## **Introduction**

**CoolSpools Email** (formerly called “**Communiqué**”) provides IBM System i users and applications with easy to use but fully functioned facilities for sending email messages in MIME format.

With **CoolSpools Email** you can:

- send email to Internet email users as well as email users on the same IBM System i or another IBM System i
- send email messages in simple text or HTML or alternative text/HTML formats
- attach or embed text, HTML, image, video, audio or application binary files (for example, PDF files and Excel spreadsheets)
- use a simple command interface or an API
- create and manage named email address lists
- send to multiple email addresses or to named email address lists
- convert SNADS distribution lists to email address lists

**CoolSpools Email** also dovetails with the rest of the **CoolSpools** suite, enabling **CoolSpools Spool Converter** to email spooled files as PDFs , XML etc. and **CoolSpools Database** to email database and query output as Excel spreadsheets, CSVs etc.

## **Why Choose CoolSpools Email?**

There are many PC packages which enable you to send email. There are also packages that let you to send email from your IBM System i using a GUI. So why choose **CoolSpools Email**?

**CoolSpools Email** provides a means of sending email from an IBM System i command. That may not be as pretty as a GUI, but it's a darn sight easier to integrate with your existing applications.

Just think about it. How is a GUI going to help you convert your billing application to let you send invoices out to customers by email? With **CoolSpools Spool Converter** and **CoolSpools Email** it can take little more than adding a few command calls to your CL program to convert your invoice spooled files to PDF format and email them.

To take another example, what if you want to email your internal users their sales report in Excel format each morning rather than printing it and distributing it on paper? A GUI is not much help there. With **CoolSpools Database** and **CoolSpools Email**, you can run your query and save the output as an Excel file rather than a spooled file and then email the file as an attachment, all with a couple of simple program changes to call the appropriate commands.

One final idea for you: do you have to support IBM System i batch jobs that run overnight or at weekends? Why not put a call to **CoolSpools Email** in your program error handling routines? That way, when an error occurs, you can receive an email at home or in the office. Neat.

## **Prerequisites**

OS/400 V5R3M0 or higher.

20 Mb of IBM System i disk space.

No PC is required.

## Upgrade Notes

Please read the following notes carefully before upgrading to CoolSpools Email V6R1M0 from an earlier version of Communiqué or CoolSpools PLUS.

CoolSpools Email was previously available as both a standalone module called Communiqué and also as part of the complete CoolSpools PLUS suite. Now, CoolSpools Email is packaged as a product option (option 2) of CoolSpools V6R1M0.

To determine which version of CoolSpools PLUS or Communiqué you are running, check the name of the library in which the SNDCMNMSG command object you are using resides, e.g.:

### **DSPOBJD OBJ(SNDCMNMSG) OBJTYPE(\*CMD)**

The library name corresponds to the product and version of CoolSpools as shown in the table below:

<b>Product Library Name</b>	<b>Licensed Program Id</b>	<b>Version</b>
CMNQUEV1R1	1CMNQUE	Communiqué Version 1
COOLSPV5R1	5COOLSP	CoolSpools PLUS V5R1M0

If your command is in a library other than those shown, you are either running an unsupported, unlicensed version or have moved or copied the command object from its original location. Contact [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk) for assistance.

## License Keys

You are entitled to upgrade to V6R1M0 of CoolSpools Email free of charge if:

- the machine on which you wish to run CoolSpools Email V6R1M0 has a valid license for an earlier version of Communiqué or CoolSpools PLUS

**and**

- you are either in your first 12 months' maintenance period after purchase or have paid your latest annual maintenance invoice.

If you wish to upgrade, you can simply download the software from [www.ariadnesoftware.co.uk](http://www.ariadnesoftware.co.uk) and install it according to the instructions contained in the

“Installation” section of this User Guide. However, if you licensed an earlier version of Communiqué or CoolSpools PLUS, you will need to request a license key for the new version. Simply e-mail [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk) and ask for your key for V6R1M0 of CoolSpools Email. Please quote your system serial number(s) and processor group code(s) in your e-mail. These are shown at the top of the WRKLICINF screen.

Without a license key, CoolSpools Email V6R1M0 will allow you a 30-day grace period and will then no longer run.

If you have not paid your annual maintenance invoice, and if you need longer than 30-days to test CoolSpools Email V6R1M0, we will, on request, send you a temporary license key to extend the trial period.

If you require additional temporary license keys to assist with testing CoolSpools Email V6R1M0, or if you run into any problems during your testing, please do not hesitate to contact us at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

### **Warning/Disclaimer**

**We recommend strongly that all production applications are re-tested thoroughly using the new version in your development environment before you switch over to running the new version in your production environment.**

All CoolSpools versions are packaged as separate licensed programs and install into different libraries. This means that all versions of CoolSpools can coexist and run alongside one another on the same machine. You can switch an application from using one version to using another simply by changing the library list of the job to include the appropriate version library or by specifying a different library name when you run the command. Hence it is quite a simple matter to test your applications using the new version while continuing to run the older version in production.

Please note that while ariadne makes every effort to ensure that CoolSpools functions in the same way with the same parameters from one version to the next, it is not possible to guarantee this. This is why you should re-test your applications against a new version before going live with it as it is possible that in some cases different parameter settings will be necessary to obtain the same results as before.

ariadne software accepts no responsibility for any damage, expense or loss of income incurred as a result of unforeseen and unwanted effects resulting from installing new versions of its software or applying PTFs.

### **Minimum OS/400 Release Level**

The minimum OS/400 release level required to run V6R1M0 of CoolSpools Spool Converter is OS/400 **V5R3M0**.

If you are running OS/400 V5R2M0 or an earlier version of OS400, you will not be able to install V6R1M0 of CoolSpools Email.

## **Product Library**

All product options of CoolSpools V6R1M0 install into the single product library **COOLSPV6R1**. This means that you no longer have to manage multiple product libraries for the separate modules that made up CoolSpools PLUS V5R1M0 (CoolSpools Database, Communiqué, CoolSpools etc.)

You will probably need to change library lists in job descriptions and other system objects in order to pick up the new version of the code rather than the old.

This change of library name has the advantage that it allows you to run both V6R1M0 and the earlier versions on the same machine. You are therefore able to test V6R1M0 before swapping your production applications over to the new version, as we strongly advise you to do.

## **Memo to Users**

Please refer to the [Memo to Users](#) for important information about changes you need to take account of before migrating to CoolSpools Email from an earlier version of Communiqué or CoolSpools PLUS.

## **What's new in this release?**

- **SNDCMNMSG command enhancements**

New SAVE parameter allows the messages to be saved for re-sending at a later time. Saved messages can be encrypted and/or compressed.

New LOG parameter lets you control whether a message is logged and whether the logged information is encrypted.

***N.B. CoolSpools Email does not currently provide support for encrypted messages. This encryption relates to information stored in the message log and saved messages.***

New PWD parameter allows you to supply an encryption password which needs to be supplied before a previously saved, encrypted password can be retrieved.

New EMAILID parameter allows you to retrieve a previously saved email message and send a new message based on the previous message. Attachments are re-attached and email parameters can be modified.

- **New CRTCMNMSG command**

This new command provides a more user-friendly interface for sending emails than the SNDCMNMSG command line interface.

- **New DLTCMNMSG command**

This new command provides a way of deleting saved messages and message log contents based on the age of the message.

- **New RSNCMNMSG command**

This new command lets you re-send a saved message exactly as it was originally saved (with the original attachments intact).

- **New SNDCMNVCAL and SNDCMNICAL commands**

These new commands let you send calendar entries (appointments, meeting invitations etc.) which can be received by any calendar application such as Outlook which supports vCalendar and iCalendar format messages.

- **New API options**

(Some of these options were added to V1 by PTF since the original release).

New alternative API format supports:

- Message texts greater than 4096 bytes (no practical imposed maximum length).
- Up to 1024 attachments and recipients possible (original maximum 32)
- Message text supplied in a stream file or database file member
- Confirmation of delivery sent to a different email address or addresses from the sender's email address
- Specification of options for saving email messages
- Specification of options for logging email messages
- Passwords for saved and/or logged email messages when these are encrypted

## **Important Note about POP settings!**

Prior to OS/400 V5R3M0, the default for the POP MSGSPLIT attribute was set to a value which meant that large email attachments were split across multiple messages. This often resulted in those email attachments becoming corrupted by the AS/400 email servers after they had been created by CoolSpools Email.

We recommend strongly that you set your MSGSPLIT level to \*NOMAX to prevent this happening, e.g.:

```
CHGPOPA MSGSPLIT(*NOMAX)
```

If you are experiencing problems with large attachments getting corrupted, this is almost certainly the solution.

## **Address Lists**

**CoolSpools Email** implements the concept of a named **Email Address List**. Address lists can be extremely useful when you need to send messages to groups of people on a regular basis. For example, you might create an address list for each department within your organization (e.g. SALES, MARKETING, IT etc.). Then, when you need to send a message to everyone in the Sales Dept., you just specify the **SALES** address list as a recipient for the message.

**CoolSpools Email** provides a set of tools for creating and managing address lists and their contents.

## **Address Directories**

**CoolSpools Email** also implements the concept of a named **Email Address Directory**. Address directories store frequently used email addresses. You can search them and select email addresses from them when you specify TO(\*SELECT) on the SNDCMNMSG command. Different directories can be used to keep email addresses relevant to different users or groups of users separate from one another, for ease of searching and selection. For example, you might create an address directory for each department within your organization (e.g. SALES, MARKETING, IT etc.). Then, when a member of the Sales Department needs to select an email address, s/he can simply look it up in the departmental directory.

**CoolSpools Email** provides a set of tools for creating and managing address directories and their contents.

## **Installation**

Refer to the [Installation Guide](#) for instructions.

## **Maintenance**

Refer to the [Maintenance Guide](#) for instructions.

## Setup

**CoolSpools Email** provides means of creating email messages in MIME (Multipurpose Internet Mail Extensions) format on the AS/400, iSeries or IBM System i . There is some setup work which must be carried out in order for these emails to be delivered successfully.

### How CoolSpools Email works

If you have fix pack 122 or later installed, CoolSpools Email offers two ways of sending emails from your system. If you do not yet have fix pack 122, you will only be able to use the first of these methods.

#### **1. Mail Server Framework (MSF)**

The original release of CoolSpools V6 Email, as well as the earlier *Communiqué* product, used the IBM QtmmSendMail API to create emails. QtmmSendMail sends email via two IBM-provided services for processing:

- i. **Mail Server Framework (MSF)**
- ii. IBM's **Simple Mail Transport Protocol (SMTP)** servers

For email sent this way to work, both these services need to be configured and active.

There is a section on configuring and managing MSF and the IBM SMTP servers below which links to a document which also includes some tips on how to obtain diagnostic information and trace problems.

Option 41 from the EMAIL menu "Manage CoolSpools email sent via IBM's MSF" provides access to a number of functions for setting up and managing this method of delivering emails. You can also access that menu by entering the command:

#### **GO COOLSPV6R1/MSFEMAIL**

While this is the method of sending emails that many customers have used very successfully for years, this method of sending emails from the system i has a number of drawbacks, which IBM seems disinclined to address:

- a) It can be complex to set up and manage
- b) Diagnostic and logging information can be hard to obtain and interpret
- c) Tracing an email from the point at which it is created through to the point at which it is (hopefully) delivered or (sometimes) rejected, can be tough (or even impossible)

In consequence, starting with fix pack 122, CoolSpools Email now also fully supports an alternative method of sending emails from your system i and we have made it very easy to switch over to this alternative method without having to alter your applications in any way.

## ***2. CoolSpools SMTP***

The new CoolSpools SMTP method doesn't use either of the IBM services described above to send emails. Instead, it uses CoolSpools own implementation of the Simple Mail Transport Protocol (SMTP) to communicate directly with the target email system to deliver emails.

This has allowed us to address the issues described above, namely:

- a) We think it's easier to set up and manage
- b) We believe the diagnostic and logging information provided is much improved
- c) We can now take charge of the entire process of sending an email from beginning to end and give you the ability to tell very easily whether a particular email has been successfully sent or not, and if not, why not.

Option 42 from the EMAIL menu “Manage CoolSpools email sent via CoolSpools SMTP” provides access to a number of functions for setting up and managing this method of delivering emails. You can also access that menu by entering the command:

**GO COOLSPV6R1/SMTPEMAIL**

### ***Telling CoolSpools Email which method to use***

This is very simple.

Assuming fix pack 122 is installed, there is a new option controlling the method which is used on the following commands:

- CoolSpools Email SNDCMNMSG
  - METHOD parameter
  - “MSF or SMTP” element
- CoolSpools Spool Converter CVTSPLPDF, CVTSPLTXT etc.
  - EMAILOPT parameter
  - “MSF or SMTP” element
- CoolSpools Database CVTDBFXL, CVTDBFTXT etc.

- EMAILOPT parameter
- “MSF or SMTP” element

In each case the default is \*ENVVAR, which means that, by default, the system will use the method identified by an environment variable called CS\_EMAIL\_METHOD. If you set this at system level to either \*MSF or \*SMTP, the Mail Server Framework or CoolSpools SMTP method becomes the default system-wide, respectively. If the environment variable does not exist, or if it has a value other than \*MSF or \*SMTP, Mail Server Framework is currently the default, as before fix pack 122.

To set this environment variable, you can either run one of these commands:

```
ADDENVVAR
  ENVVAR(CS_EMAIL_METHOD)
  VALUE(*MSF)
  LEVEL(*SYS)
```

```
ADDENVVAR
  ENVVAR(CS_EMAIL_METHOD)
  VALUE(*SMTP)
  LEVEL(*SYS)
```

or take one of the new options from the EMAIL menu:

**43. Change default email delivery method to IBM MSF**

**44. Change default email delivery method to CoolSpools SMTP**

which do the same thing.

This will change the default method used for sending emails, but only in respect of emails sent by CoolSpools Version 6 - this doesn't affect any other AS/400 applications that send email including earlier versions of CoolSpools/Communiqué.

If, for some reason, you want to use a different method for a particular email (perhaps to test the new \*SMTP method while continuing to use \*MSF as the default), you can specify the method explicitly on the appropriate command parameter also (METHOD parameter of SNDCMNMSG or EMAILOPT parameter of CVTSPLxxx or CVTDBFxxx). The options are \*MSF (Mail Server Framework) or \*SMTP (CoolSpools SMTP).

## **[Configuring CoolSpools Email to use Mail Server Framework](#)**

See the Email Setup guide <http://www.ariadnesoftware.co.uk/docs/EmailSetup.pdf>

For further information on how to set up your IBM System i for email, consult the following IBM links:

<http://publib.boulder.ibm.com/infocenter/iseriess/v7r1m0/topic/rzair/rzairconfigem.htm>

[http://www-912.ibm.com/s\\_dir/SLKBase.nsf/3ff1161e61e6c7468625661300765a3f/34879f5e413aa5cd8625697b0053f36a](http://www-912.ibm.com/s_dir/SLKBase.nsf/3ff1161e61e6c7468625661300765a3f/34879f5e413aa5cd8625697b0053f36a)

## **Configuring CoolSpools Email to use CoolSpools SMTP**

Before selecting CoolSpools SMTP as the method to be used to deliver your emails (see above), you will need to do a small amount of simple setup work so CoolSpools SMTP knows how to run on your system.

By default, emails sent via the \*SMTP method are processed asynchronously by a number of server jobs that run in batch mode. In other words, when you run the SNDCMNMSG command, or one of the CVTSPLxxx or CVTDBFxxx commands, those commands just pass the email to the CoolSpools SMTP server jobs for sending and the actual process of sending the email is done independently. This has several advantages:

- The sending job doesn't "block" while the email is being delivered to the target email server.
- If there's a problem sending the email (e.g. a communications outage or the target email server being down temporarily), CoolSpools SMTP can automatically retry sending the email later.

To set up these CoolSpools SMTP server jobs, run **?CFGSMTPSVR** (Configure SMTP Server) to prompt this command or do **GO COOLSPV6R1/EMAIL** to display the Email menu and take option 42. **Manage CoolSpools email sent CoolSpools SMTP.**

The CFGSMTPSVR command is described in detail below. Most users will need to change no more than one or two parameters from the defaults supplied and this step only needs to be carried out once, unless there is a need to adjust the configuration subsequently.

## **CFGSMTPSVR (Configure SMTP Server) command**

This command configures the CoolSpools SMTP Server jobs which manage the process of sending emails when the \*SMTP method is selected (see above).

It is available as option 1 of the SMTPEMAIL menu.

### **Restrictions:**

- You must have \*SECADM and \*IOSYSCFG special authority to run this command.
- You must have \*CHANGE authority to the objects referenced, e.g. the subsystem, job queue and user profile specified, as well as to the CoolSpools SMTP Server

and Retry Server job descriptions (objects CM\_SNDSVRR and CM\_RTYSVRR \*JOB in library COOLSPV6R1).

Command parameters are as follows:

### ***Autostart servers (AUTOSTART)***

Specifies whether the CoolSpools SMTP server jobs should start automatically or not.

Emails sent using the CoolSpools SMTP method will not be processed unless the CoolSpools SMTP server jobs are started. If you intend to use this method to deliver emails, it is therefore recommended that you specify AUTOSTART(\*YES).

If you specify AUTOSTART(\*YES), CoolSpools will add two prestart job entries to the subsystem nominated below:

- Program CM\_SNDSVRR (Send Server - this does the sending)
- Program CM\_RTYSVRR (Retry Server - this manages retries)

When the nominated subsystem starts, these prestart job entries will cause the required jobs to start.

Options are:

<b><u>*SAME</u></b>	No change is made to the existing setting.
<b>*YES</b>	The server jobs will start automatically.
<b>*NO</b>	The server jobs will not start automatically. You will need to start them manually with the STRSMTPSVR command.

### ***Number of initial servers (NBRSVR)***

Specifies how many SMTP server jobs should be started initially when the servers are set to AUTOSTART(\*YES).

The number of servers that should be started will depend on the volume of emails being generated. If a large volume of emails is being sent, and a backlog is developing because the number of server jobs available to process those emails cannot cope with that volume, starting additional server jobs will probably improve throughput.

The default is 3, which will be a reasonable number for most environments.

Additional servers can be started with the STRSMTPSVR command.

Options are:

**\*SAME** No change is made to the existing setting.

**1-99** Specify the number of server jobs to autostart.

### ***Run as user profile (USER)***

Specifies the user profile under which the SMTP server jobs will run. The user profile must have sufficient authority to all required objects, in particular \*USE authority to library COOLSPV6R1 and the programs and files it contains.

The default is QUSER. There is no need to change this unless QUSER does not have sufficient authority to play this role.

Options are:

**\*SAME** No change is made to the existing setting.

**\*JOB** The user profile specified in the job description is used. The job description is CM\_SNDVRR for the Send Server and CM\_RTYSVRR for the Retry Server.

**name** Specify the name of the user profile.

### ***Data queue library (DTAQLIB)***

Specifies the name of the library in which the data queue used to communicate between the sending application and the CoolSpools SMTP server jobs will be created.

The default is \*PRDLIB, which means that the data queue will be created in the product library COOLSPV6R1. There is no need to change this default unless there is some reason why that library is inappropriate.

Options are:

**\*SAME** No change is made to the existing setting.

**\*PRDLIB** The data queue is created in the product library (COOLSPV6R1).

**name** Specify the name of the library where the data queue will be created.

### ***Write log records (LOG)***

Specifies whether or not log records should be written for all emails.

It is recommended that this option be left as the default \*YES unless the volume of data being written becomes a problem.

Options are:

<b><u>*SAME</u></b>	No change is made to the existing setting.
<b>*YES</b>	Logging information is written for all emails, successful or unsuccessful.
<b>*NO</b>	Only error information is logged.

### ***Report errors to message queue (ERRMSGQ)***

Specifies the name of a message queue to which CoolSpools will send a message when an error occurs in the CoolSpools SMTP server jobs.

The default is the System Operator message queue QSYSOPR.

Single values

<b><u>*SAME</u></b>	No change is made to the existing setting.
<b>*NONE</b>	No message is sent.

Other options are:

Qualifier 1: Report errors to message queue

<b>QSYSOPR</b>	The message is sent to the System Operator message queue
<b>name</b>	Specify the name of name of the message queue.

Qualifier 2: Library

<b>*LIBL</b>	The message queue is located by means of the library list.
<b>*CURLIB</b>	The message queue is located in the current library of the job.
<b>name</b>	Specify the name of library where the message queue is located.

### ***Report errors to email address (ERREMAIL)***

Specifies an email address to which CoolSpools will send a message when an error occurs in the CoolSpools SMTP server jobs.

Options are:

<b>*SAME</b>	No change is made to the existing setting.
<b>*NONE</b>	No message is sent.
<b>address</b>	Specify the email address to which information about errors should be sent.

### ***Report non-delivery to msgq (DSNMSGQ)***

Specifies the name of a message queue to which CoolSpools will send a message when an email cannot be delivered.

Single values

<b>*SAME</b>	No change is made to the existing setting.
<b>*NONE</b>	No message is sent.

Other options are:

Qualifier 1: Report non-delivery to msgq

<b>QSYSOPR</b>	The message is sent to the System Operator message queue
<b>name</b>	Specify the name of name of the message queue.

Qualifier 2: Library

<b>*LIBL</b>	The message queue is located by means of the library list.
<b>*CURLIB</b>	The message queue is located in the current library of the job.
<b>name</b>	Specify the name of library where the message queue is located.

### ***Report non-delivery to email (DSNEMAIL)***

Specifies an email address to which CoolSpools will send a message when an email cannot be delivered.

It is strongly recommended you supply an email address on this parameter and that someone in your organization is given the task of monitoring this email address regularly.

This will then ensure that any emails which cannot be sent, for whatever reason, can be corrected in the appropriate fashion.

You must always anticipate that some emails that you send will not be delivered, because, for example:

- It is very easy to mis-key an email address as you enter it. Although CoolSpools email validates email addresses, what you have entered may be “well formed”, i.e. look like a valid email address, but still not be usable
- The email address you’re sending to may have been valid at some point in the past but may no longer be valid. For example, if you’re emailing a customer, the person concerned might have left the company or changed their name (e.g. got married) or the company could have changed its name or been taken over. All of these things can invalidate email addresses that worked at one time.

Options are:

<b><u>*SAME</u></b>	No change is made to the existing setting.
<b>*NONE</b>	No message is sent.
<b>address</b>	Specify the email address to which information about non-delivery should be sent.

### ***Autostart in subsystem (SBSD)***

Specifies the name of the subsystem in which CoolSpools SMTP server jobs will autostart.

This is the subsystem to which CoolSpools will add two prestart job entries if AUTOSTART(\*YES) is specified.

It is still your responsibility to ensure that this subsystem is started when the system starts, e.g. as part of the system startup program.

The default is the QUSRWRK subsystem. This will be a convenient subsystem for most installations, but, if there is some reason why you cannot use QUSRWRK, you can nominate a different subsystem here.

If you have CoolSpools Spool Admin installed, you could choose to use the **COOLSPOLS** subsystem supplied with that product option.

Single values

<b><u>*SAME</u></b>	No change is made to the existing setting.
---------------------	--

Other options are:

Qualifier 1: Autostart in subsystem

**QUSRWRK** The default subsystem is QUSRWRK.  
**name** Specify the name of the subsystem description.

Qualifier 2: Library

**\*LIBL** The subsystem description is located by means of the library list.  
**\*CURLIB** The subsystem description is located in the current library.  
**name** Specify the name of the library in which the subsystem description is located.

## ***Class (CLS)***

Specifies the class associated with the prestart job entry.

The class determines certain attributes of the job, such as its priority and timeslice.

You are unlikely to need to change this option, but can do so if you need to adjust the performance of the servers by modifying things like priority and timeslice.

Single values

**\*SAME** No change is made to the existing setting.

Other options are:

Qualifier 1: Class

**QBATCH** The QBATCH class is used. If this gives too low a priority and performance is not adequate, specify an alternative class with a higher priority level.  
**name** Specify the name of the class.

Qualifier 2: Library

**\*LIBL** Locate the class through the library list.  
**\*CURLIB** The class is located in the current library.  
**name** Specify the name of the library where the class is located.

## **Submit to job queue (JOBQ)**

Specifies the name of the job queue to which the CoolSpools SMTP server jobs are submitted when they are started manually using the STRSMTPSVR command.

You are unlikely to need to change this option unless you have specified a subsystem other than QUSRWRK for the SBSDB parameter above, in which case you should specify the name of a suitable job queue which has a job queue entry in the nominated subsystem here.

It is your responsibility to ensure that the job queue specified has a suitable job queue entry in the subsystem description of the subsystem in which you wish these jobs to run. The job queue entry must allow at least the required number of send server jobs plus the one retry server job to run simultaneously.

Single values

**\*SAME** No change is made to the existing setting.

Other options are:

Qualifier 1: Submit to job queue

**QUSRNOMAX** The system-supplied job queue QUSRNOMAX is used. As supplied by IBM, this has a job queue entry in subsystem QUSRWRK which allows an unlimited number of jobs to run concurrently.

**\*JOBQ** The job queue specified in the job description CM\_SNDVRR (Send Server) or CM\_RTYSVRR (Retry Server) is used.

**name** Specify the name of the job queue.

Qualifier 2: Library

**\*LIBL** The job queue is located through the library list.

**\*CURLIB** The job queue is located in the current library.

**name** Specify the name of the library in which the job queue is located.

## **First-level retries (LVL1RETRY)**

When an email cannot be sent, CoolSpools will:

- First retry sending the email the specified maximum number of first-level retries,

waiting the specified number of *minutes* between retries.

- Then retry sending the email the specified maximum number of second-level retries, waiting the specified number of *hours* between retries.
- Finally, retry sending the email the specified maximum number of third-level retries, waiting the specified number of *days* between retries.

Where the email could not be sent due to a temporary outage, it is possible a subsequent retry will succeed.

Single values

**\*SAME** No change is made to the existing setting.

**\*NONE** No first-level retries will be made.

Other options are:

Element 1: Number of attempts

**1-99** Specify the number of first-level retries to attempt.

Element 2: Retry interval (minutes)

**1-60** Specify the number of minutes to wait between retries.

### ***Second-level retries (LVL2RETRY)***

Specifies the number of second-level retries to be made and the interval in hours between second-level retries.

Single values

**\*SAME** No change is made to the existing setting.

**\*NONE** No second-level retries will be made.

Other options are:

Element 1: Number of attempts

**1-99** Specify the number of second-level retries to attempt.

Element 2: Retry interval (hours)

**1-60** Specify the number of hours to wait between retries.

### ***Third-level retries (LVL3RETRY)***

Specifies the number of third-level retries to be made and the interval in days between third-level retries.

Single values

**\*SAME** No change is made to the existing setting.

**\*NONE** No third-level retries will be made.

Other options are:

Element 1: Number of attempts

**1-99** Specify the number of third-level retries to attempt.

Element 2: Retry interval (hours)

**1-60** Specify the number of days to wait between retries.

### ***Host name (HOSTNAME)***

Specifies the host name to use when identifying the local system to a target email server.

It is unlikely you will need to change this value unless the host name specified in the TCP/IP attributes is invalid and cannot be corrected.

Options are:

**\*SAME** No change is made to the existing setting.

**\*TCPATR** The host name specified in the local system's TCP/IP attributes is used (CHGTCPDMN command).

**name** Specify the host name to use.

### ***Domain name (DMNNAME)***

Specifies the domain name to use when identifying the local system to a target email server.

It is unlikely you will need to change this value unless the host name specified in the TCP/IP attributes is invalid and cannot be corrected.

Options are:

**\*SAME** No change is made to the existing setting.

**\*TCPATR** The domain name specified in the local system's TCP/IP attributes is used (CHGTCPDMN command).

**name** Specify the domain name to use.

### ***Mail server name or IP address (MAILSERVER)***

Specifies the name or IP address of a mail server via which emails should be sent.

Where it is not required or possible to send emails directly to the target email servers, and emails are to be routed via a mail server or router, specify the name or IP address of that mail server or router here.

You must ensure that the mail server or router has been configured to allow emails from the local system to be relayed through it.

Where a mail server name is specified, you must ensure that the local system can translate that name into an IP address either by means of a Domain Name Server (DNS) server or a local host table entry (CFGTCP option 10).

Options are:

**\*SAME** No change is made to the existing setting.

**\*NONE** No mail server or router is to be used, Emails will be delivered by communicating directly with the target email server.

**name** Specify the name or IP address of the mail server or router.

### ***Server authentication (SVRAUTH)***

Specifies the authentication method to be used when communicating when the mail server.

#### **Note about the use of TLS (Transport Layer Security)**

Some of the authentication methods listed below (those with names starting \*TLS) require the use of Transport Layer Security (TLS).

The use of TLS to encrypt the interactions between the sending system and receiving system significantly improves the security of your communications. However, please note the following prerequisites for using TLS:

- The mail server must support the STARTTLS verb. Contact the administrator of your mail server to verify this.

- Your system must have a valid SSL certificate associated with the CoolSpools SMTP application. You can use IBM's Digital Certificate Manager software to associate a certificate with an application. The name of the application that requires the certificate associated with it is ARIADNE\_SMTP\_SERVER

#### Single values

**\*SAME** The authentication mechanism is not changed.

**\*NONE** No authentication is used.

#### Element 1: Mechanism

**\*LOGIN** The AUTH LOGIN mechanism is used. This requires a user ID and password to be provided to the server. The user ID and password are not sent as plain text but the level of encryption used is very weak.

**\*PLAIN** The AUTH PLAIN mechanism is used. This requires a user ID and password to be provided to the server. The user ID and password are not sent as plain text but the level of encryption used is very weak.

**\*CRAMMD5** The AUTH CRAM-MD5 mechanism is used. This requires a user ID and password to be provided to the server. The password is never itself sent across the connection and the level of encryption used is strong.

**\*TLSLOGIN** The AUTH LOGIN mechanism is used. This requires a user ID and password to be provided to the server.

Transport Layer Security (TLS) is used to encrypt traffic between the sending system and the receiving system, making this mechanism significantly more secure than \*LOGIN.

**\*TLSPLAIN** The AUTH PLAIN mechanism is used. This requires a user ID and password to be provided to the server.

Transport Layer Security (TLS) is used to encrypt traffic between the sending system and the receiving system, making this mechanism significantly more secure than \*PLAIN.

**\*TLSCRAM** The AUTH CRAM-MD5 mechanism is used. This requires a user ID and password to be provided to the server. The password is never itself sent across the connection and the level of encryption used is strong.

Transport Layer Security (TLS) is used to encrypt traffic between the sending system and the receiving system, making this mechanism significantly more secure than \*CRAM-MD5 alone.

### ***Server login (SVRLOGIN)***

Specifies the user ID and password that are used for authentication purposes.

Single values

- \*SAME**      The information is unchanged.
- \*NONE**      No user ID or password is provided.

Element 1: User

- char-value**      Specify the user ID.

Element 2: Password

- char-value**      Specify the user password

## **Managing the CoolSpools SMTP servers**

When you run the Configure SMTP Server command, if you specified \*YES for the **Autostart servers** options, CoolSpools will attempt to set up your environment ready for using this method of delivering emails. Assuming no problems occurred (e.g. no authority issues arose), it will have added the necessary prestart job entries to the subsystem you chose. If that subsystem is active, the CoolSpools SMTP server jobs should automatically start to run. To check, do:

```
WRKSBSJOB SBS(QUSRWRK)
```

or

```
WRKSBSJOB SBS(subsystem_name_here)
```

(if you nominated a different subsystem) and see if you can find jobs called CM\_SNDVRR and CM\_RTYSVRR. If they are there, then CoolSpools was able to add and start the prestart job entries and you're ready to send emails this way.

If at any time you need to start or end the CoolSpools SMTP servers manually, you can use the STRSMTPSVR (Start SMTP Servers) and ENDSMTPSVR (End SMTP Server) commands to do so.

### **STRSMTPSVR (Start SMTP Server) command**

The Start CoolSpools SMTP Server (STRSMTPSVR) command starts the specified number of CoolSpools SMTP server jobs used to send emails via the CoolSpools SMTP server method.

It is available as option 11 of the SMTPEMAIL menu.

#### **Restrictions**

- You must have \*JOBCTL special authority to run this command.
- The jobs will be submitted to the job queue specified on the JOBQ option of the CoolSpools SMTP server attributes (CFGSMTPSVR command) is started.
- You must ensure that this job queue has a job queue entry in an active subsystem and that at least the number of send server jobs specified can run concurrently, plus the retry server job.

### **Number of server jobs to start (NBRJOBS)**

Specifies the number of jobs used for sending emails that will be started.

Options are:

- |                |   |
|----------------|---|
| <b>*SVRATR</b> | The number of server jobs specified on the CFGSMTPSVR NBRSVR (Number of initial servers) option of the CoolSpools SMTP server attributes (CFGSMTPSVR command) is started. |
| <b>1-999</b>   | Specify the number of send server jobs to start.  |

### ***Start retry server if inactive (RTYSVR)***

Specifies whether or not to start a retry server job if one is not currently active.

The retry server job manages the retrying of emails that could not be sent due to an error.

Options are:

- |             |   |
|-------------|---|
| <b>*YES</b> | A retry server job is started if one is not already active. |
| <b>*NO</b>  | No retry server job is started.                             |

### **ENDSMTPSVR (End SMTP Server) command**

The End CoolSpools SMTP Server (ENDSMTPSVR) command instructs the CoolSpools SMTP server jobs and the Retry Server to end.

A control message is placed on the data queue and will be detected as soon as the jobs have finished whatever work they are currently performing. There could therefore be a slight delay before the jobs actually end.

It is available as option 12 of the SMTPEMAIL menu.

#### **Restrictions**

- You must have \*JOBCTL special authority to run this command.

There are no parameters.

### **WRKSMTPMSG (Work with SMTP Messages) command**

The Work with SMTP Messages (WRKSMTPMSG) command lets you work with email messages sent via the CoolSpools SMTP server.

This option demonstrates some of the advantages of the CoolSpools SMTP method of sending emails over the Mail Server Framework method. It provides a single place where you can:

- track what has happened to the emails you have sent, in terms of their current status and diagnostic information about any errors that occurred
- resolve issues such as invalid email address and trigger re-sending of the email, or create new emails based on old ones

It is available as option 21-23 of the SMTPEMAIL menu.

21. Work with CoolSpools SMTP emails - unprocessed

22. Work with CoolSpools SMTP emails - in error

23. Work with CoolSpools SMTP emails - all statuses

### Restrictions:

- Unless you are a system administrator, you can only work with emails that you sent. In order to be considered a system administrator, you must either:
  - Have \*SECADM special authority; or
  - Be authorized to registered function ARIADNE\_SYS\_ADMIN. Use the CHGFCNUSG or WRKREGFNC commands to manage authority to registered functions.
- System administrators are permitted to view and work with (e.g. re-send) emails sent by any user.

### **Option (OPTION)**

Specifies the status that messages must have to be displayed.

Options are:

<b><u>*UNPRC</u></b>	Displays unprocessed messages.  The list will include all messages that could not be delivered, due to an error, as well as those for which delivery has not yet been attempted and which are waiting to be processed.
<b>*ERROR</b>	Displays messages in error.  The list will include just those messages for which delivery has been attempted at least once but where delivery has not yet been processed successfully.

**\*ALL** All messages are shown, irrespective of their status.

Once the list of emails has been displayed, refer to the online help text (available by pressing F1) for information about the information displayed and the options available.

# Using CoolSpools Email

## The CoolSpools Email Menu

The easiest way to get started with **CoolSpools Email** is to use **CoolSpools Email** menu. The **CoolSpools Email** menu brings together all of **CoolSpools Email's** command-line options in a single convenient place.

To display the **CoolSpools Email** menu, enter the following command at an AS/400 command line:

```
GO COOLSPV6R1/EMAIL
```

```
-----  
EMAIL                               CoolSpools Email Menu  
  
Select one of the following:  
  
    1. Send email message             2. Create email message  
    3. Retrieve email message         4. Re-send email message  
  
    10. Work with email address lists  11. Convert distribution list  
  
    21. Work with email addr directories  
  
    31. Send iCalendar item           32. Send vCalendar item  
  
    41. Manage CoolSpools email sent via IBM's MSF  
    42. Manage CoolSpools email sent via CoolSpools SMTP  
    43. Change default email delivery method to IBM MSF  
    44. Change default email delivery method to CoolSpools SMTP
```

```
Selection or command  
===>  
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel  
F13=Information Assistant   F16=AS/400 main menu
```

-----  
Some of these options were described above. In the following sections we will explore the various other options available from this menu.

## **SNDCMNMSG command**

Option 1 (Send email message) and option 3 (Retrieve email message) from the **CoolSpools Email** menu prompt the **SNDCMNMSG** (Send CoolSpools Email Message) command which allows you to send an email message.

The same functionality is available from option 2 (the CRTCMNMSG (Create CoolSpools Email Message)) command and the **CoolSpools Email** Send Message API (Application Program Interface), which are described in a later section of this document. CRTCMNMSG provides a more user-friendly interface for creating email messages.

The difference between options 1 and 3 is that option 2 supplies the default value for the EMAILID parameter (EMAILID(\*NEW)) and prevents this parameter from being changed. This is more conveniently if you are sure you just want to create a brand new email. Option 3, on the other hand, allows you to input the identifier of a previously saved email message on the EMAILID parameter in order to retrieve that message as the basis for the new message.

Command parameters are as follows:

### ***EMAILID - Email id to retrieve***

The **EMAILID** (Email id to retrieve) parameter allows you to specify the identifier of a previously saved email to be used as the basis for the new message.

Options for the first element (Email id) are:

<b>*NEW</b>	(Default) No previously saved email message is retrieved.
<b>Email_id</b>	The identifier of the previously saved email to be retrieved as the basis for the new message.

If the email that is to be retrieved was encrypted when it was saved, you must supply the encryption password on the second element ("Encryption password").

### ***TO - Recipient(s)***

The **TO** (Recipients) parameter allows you to specify the email addresses to which the email message should be sent.

You can define up to 32 recipients for the message on this command parameter. If you need to send the same email address to more than 32 recipients simultaneously, you can do this by defining an address list and specifying the address list name on this parameter.

The default is the single value **\*SELECT**: CoolSpools Email will prompt you to enter the

email address(es) to which the message should be sent or select the address(es) from an email address directory.

There are 3 elements to this parameter: **Email address**, **Name** and **Type**.

### **Email address**

This is where you enter the email address to which the message is to be sent.

Note that while **CoolSpools Email** will check that the email address that you enter conforms to the rules for valid email addresses, it is not possible to validate that the email address that you enter is correct or that the message will be deliverable.

For example, [sales.ariadnesoftware.co.uk](mailto:sales.ariadnesoftware.co.uk) is not a valid email address (since it does not contain an @ sign), and **CoolSpools Email** will reject it. However, [sales@ariadnesoftware.org.uk](mailto:sales@ariadnesoftware.org.uk) is a valid email address and **CoolSpools Email** will allow it, but it is not ariadne's correct email address (it should be [sales@ariadnesoftware.co.uk](mailto:sales@ariadnesoftware.co.uk)) and the message will not be received.

### **Name**

If you would like your email message to display the recipient's name rather than the email address when it is delivered, enter the name here.

The default value is \*NONE, i.e. no name is provided and the email address will appear as the recipient instead.

For example, if you specify:

```
SNDCMNMSG TO((Sales@ariadnesoftware.co.uk *NONE))
```

when the message is received, the To: attribute will be shown as:

```
To: Sales@ariadnesoftware.co.uk
```

However, if you specify:

```
SNDCMNMSG TO((Sales@ariadnesoftware.co.uk 'ariadne Sales'))
```

when the message is received, the To: attribute will be shown as:

```
To: ariadne Sales
```

Options are:

- |              |   |
|--------------|---|
| <b>*NONE</b> | (Default) No name is specified.   |
| <b>*ADRL</b> | Address list. If you wish to send to an address list, this is the value that must be entered. |

**Name** Specify the name of the recipient.

### Type

Specify the type of recipient here.

Options are:

<b>*PRI</b>	(Default) Primary recipient.
<b>*CC</b>	Carbon copy recipient. An *CC recipient receives a copy of the message, and is identified to the primary recipient, but is not the primary recipient.
<b>*BCC</b>	Blind carbon copy recipient. An *BCC recipient receives a copy of the message, but is not identified to the primary recipient or *CC recipients.
<b>*ADRL</b>	Address list. If you wish to send to an address list, this is the value that must be entered.

### Example:

Sending to ariadne Sales as a primary recipient with a copy to ariadne Marketing:

```
SNDCMNMSG TO( (Sales@ariadnesoftware.co.uk 'Sales' *PRI)
               (Marketing@ariadnesoftware.co.uk 'Marketing' *CC))
```

### Example:

Sending to an email address list called "Sales":

```
SNDCMNMSG TO((Sales *ADRL *ADRL))
```

### **FROM - Sender**

The **FROM** (Sender) parameter allows you to specify the sender of the email.

The default value is **\*CURRENT**, which means that **CoolSpools Email** will try to retrieve the email address of the user sending the email from the System Distribution Directory. If no email address can be determined for the user from the System Distribution Directory, you will need to enter the values you wish to use manually.

There are 2 elements to this parameter: **Email address** and **Name**.

### **Email address**

This is where you enter the email address of the sender, which will also be the address to

which, by default, a reply to this email should be directed.

Note that while **CoolSpools Email** will check that the email address that you enter conforms to the rules for valid email addresses, it is not possible to validate that the email address that you enter is correct or that any reply sent to the message will be deliverable.

For example, [sales.ariadnesoftware.co.uk](mailto:sales.ariadnesoftware.co.uk) is not a valid email address (since it does not contain an @ sign), and **CoolSpools Email** will reject it. However, [sales@ariadnesoftware.org.uk](mailto:sales@ariadnesoftware.org.uk) is a valid email address and **CoolSpools Email** will allow it, but it is not ariadne's correct email address (it should be [sales@ariadnesoftware.co.uk](mailto:sales@ariadnesoftware.co.uk)) and any reply sent to this email address will not be received.

If **\*CURRENT** is specified, the following method is used to obtain the sender's email address:

- a) The following fields are selected from the System Distribution Directory entry of the current user (this can be changed using the WRKDIRE command).

- user id (USRID)
- user address (USRADDR)
- SMTP user id (SMTPUSRID)
- SMTP domain (SMTPDMN)

- b) We then build a sender email address using the following logic:

```
If SMTP user id and domain are non-blank
  Email address = SMTPUserId@SMTPdomain
Else
  If user id, user address, hostname and domain are all non-blank
    Email address = UserId.UserAddress@HostName.Domain
  Else
    If host name and domain are non-blank
      Email address = UserProfile@HostName.Domain
    Else
      Error CMQ9105 (Unable to identify email address for X).
    EndIf
  EndIf
EndIf
```

## Name

If you would like your email message to display a sender's name rather than the sender email address when it is delivered, enter the name here.

The default value is **\*NONE**, i.e. no name is provided and the email address will appear as the sender instead.

For example, if you specify:

```
SNDCMNMSG... FROM((Sales@ariadnesoftware.co.uk *NONE))
```

when the message is received, the **From:** attribute will be shown as:

**From:** Sales@ariadnesoftware.co.uk

However, if you specify:

**SNDCMNMSG FROM((Sales@ariadnesoftware.co.uk 'ariadne Sales'))**

when the message is received, the **From:** attribute will be shown as:

**From:** @ariadne Sales

If **\*CURRENT** is specified, the following method is used to determine the sender's name:

a) The following fields are selected from the System Distribution Directory entry of the current user (this can be changed using the WRKDIRE command).

- first name (FSTNAM)
- last name (LSTNAM)
- preferred name (PREFNAM)
- user description (USRD)
- surname (SURNAM)
- given name (GIVENNAM)
- user address (USRADDR)

b) We then build a sender name using the following logic:

```
If preferred name is non-blank
  Sender name = preferred name
Else
  If first name and last name are non-blank
    Sender name = "FirstName LastName"
  Else
    If given name and surname are non-blank
      Sender name = "GivenName Surname"
    Else
      If user description is non-blank
        Sender name = user description
      Else
        Retrieve user profile attributes
        If text description of user profile is non-blank
          Sender name = user profile text description
        Else
          Sender name = blanks (no sender name will be used and the email address
            will show as the sender)
        End If
      End If
    End If
  End If
End If
End If
End If
```

## ***REPLYTO - Send replies to***

The **REPLYTO** (Send replies to) parameter lets you specify the email address(es) to which replies to the email should be sent.

The default value is the single value **\*SENDER**, which means that replies should be sent to the email address specified on the FROM parameter.

Up to 32 email addresses can be specified.

There are 2 elements to this parameter: **Email address** and **Name**.

### **Email address**

Specify the email address to which replies should be directed or the name of an email address list or specify **\*SELECT** to select from email address directories.

### **Name**

Options are:

<b>*NONE</b>	(Default) No name is specified.
<b>*ADRL</b>	Address list. If you wish to send to an address list, this is the value that must be entered.
<b>Name</b>	Specify the name of the recipient.

## ***SUBJECT - Subject***

The **SUBJECT** (Subject) parameter allows you to define a subject line for the message. You can enter up to 256 characters of free-format text.

### **Example:**

```
SNDCMNMSG...SUBJECT('Example subject line')
```

When this message is received, it will show as:

```
Subject:    Example subject line
```

## ***MSG – Message text***

The **MSG** (Message) parameter allows you to enter the text of an email message directly on the command line.

Up to 4096 characters of free-format text can be entered here. However, due to limitations of the OS/400 command prompter, only 512 characters can be entered when the command is prompted with F4.

If you need to send more than 4096 characters of text, then your options are:

- Use MSG(\*STMF) and specify a stream file containing the text of the message to be sent on the MSGSTMF parameter.
- Use an embedded text or HTML file (see the ATTACH parameter below)
- Use the **CoolSpools Email** API, which allows virtually unlimited text if the CM\_SNDMSG2 form is used.

The message can be sent in either plain text, HTML or alternative plain text/HTML formats.

There are 2 elements to this parameter: **Message text** and **Message format**.

### **Message text:**

You can enter the text of a short message here. Up to 512 characters can be entered.

When received, the message will be displayed exactly as it is entered, with the following exceptions:

- If you want to force a line break, enter <br>. Even if the message is sent in plain text format, this HTML control will be interpreted and converted to a hard line break (carriage return-line feed sequence).
- Other HTML controls may be entered, but will only be interpreted as HTML controls if the message is sent and delivered in HTML format.

Longer messages (up to 4096 bytes) can be sent using the CoolSpools Email API CM\_SNDAPIR.

Alternatively, specify \*STMF for the message text and provide the name of a stream file on the MSGSTMF parameter. CoolSpools Email will take the text of the message from the stream file specified.

### **Message format:**

This is where you specify the format in which the message is sent.

Options are:

- \*BOTH** (Default) The message is sent in alternative plain text/HTML format. This means that two copies of the message text will be sent: a plain text copy and an HTML copy. If the email client software used to receive the message can handle HTML messages, the HTML copy will be used, otherwise the plain text copy will be used.

**\*TEXT** The message is sent in plain text format. The only HTML control which is interpreted is <br>, which **CoolSpools Email** will convert to a hard line break.

**\*HTML** The message is sent in HTML format. You can include HTML formatting (e.g. <b> </b> or <u> </u> to control bold text and underlining). *CoolSpools Email* will take the text that you enter and wrap it with some basic HTML header and footer controls (<HTML> <HEAD> <BODY>). These controls should not therefore be included in the text of the message.

**Example:**

**SNDCMNMSG...**

**MSG('Here"s a message <br>with<br>line <br>breaks.' \*TEXT)**

When this message is received, it will show as:

Here's a message  
with  
line  
breaks.

**Example:**

**SNDCMNMSG...**

**MSG( 'Here"s a demonstration of a message with HTML controls.<br>**

**<b>This line is in bold, </b><br>**

**<u>While this line is underlined.</u>')**

When this message is received, it will show as:

Here's a demonstration of a message with HTML controls.  
**This line is in bold,**  
While this line is underlined.

***ATTACH - Attachments***

The **ATTACH** (Attachments) parameter allows you to attach or embed up to 32 files to your email message.

There are 5 elements to this parameter: **File name, Method, Content type, Code page** and **Attachment name in email.**

### File name:

This is where you enter the full path name of the file you wish to attach. The path name must be entered in IFS format. In other words, documents stored in shared folders must be referenced in the QDLS file system and database file members must be reference in the QSYS.LIB file system.

### Method:

This is where you specify the method by which the file is sent.

Options are:

- |                |  |
|----------------|--|
| <b>*ATTACH</b> | (Default) The file is sent as an attachment. It will appear as an attached file separate from the text of the email.   |
| <b>*EMBED</b>  | The contents of the file are embedded in the text of the email and will follow the text of any message entered on the MSG parameter. Please note that your client email software is likely only to support the embedding of certain types of file, e.g. text, HTML and images. |

### Content type

The content type tells *CoolSpools Email* and the receiving email software what type of data the attachment represents.

It is important that the content type is specified accurately, as *CoolSpools Email* will process the attachment differently depending on the value you specify here. For example, if you specify the attachment type as \*TEXT or \*HTML, it will be handled using a technique appropriate to textual data, and may be converted from EBCDIC to ASCII, whereas any other content type will cause the attachment to be handled as binary data and sent unaltered.

Moreover, the receiving software may not handle the attachment appropriately if the content type is specified incorrectly. For example, if you send a PDF file attachment, but specify a content type of HTML, the receiving email software may attempt to display the PDF file as HTML, which will cause unwanted results.

Options are:

- |              |  |
|--------------|--|
| <b>*FILE</b> | (Default) The content type will be inferred from the extension part of the file name.<br><br>You can control which file extensions are identified as which content types by listing file extensions in the second-level text of CoolSpools Database messages |
|--------------|--|

CMQ5000-CMQ5004. Each file extension must be from 1-5 characters long, padded with trailing blanks if less than 5 characters long. For example, message CMQ5001 controls which file extensions identify image files. This message is supplied with the value:

“JPG JPEG GIF BMP PNG TIF TIFF”.

You may change the text of this message, to add or remove extensions, so long as you ensure that each extension is exactly 5 characters long (including any trailing blanks) and remembering that installing PTFs or a new release will overwrite your changes.

Any file extension not listed in one of these messages will be treated as application binary data.

<b>*BINARY</b>	Application binary. This is the default MIME content type used for data types other than those which have distinct content types of their own (image, audio, video etc.)
<b>*TEXT</b>	Identifies the attachment as plain ASCII text.
<b>*CSV</b>	Identifies the attachment as CSV or other similar delimited ASCII text.
<b>*HTML</b>	Identifies the attachment as HTML.
<b>*PDF</b>	Identifies the attachment as a PDF.
<b>*XML</b>	Identifies the attachment as XML.
<b>*IMAGE</b>	identifies the attachment as an image.
<b>*VIDEO</b>	identifies the attachment as a video file.
<b>*AUDIO</b>	identifies the attachment as an audio file.

### Code page

This element allows you to identify the code page of the data in the attachment file. This value is used if the content type is indicated as \*TEXT or \*HTML to convert the data to ASCII format so it can be displayed by the client email software.

Options are:

<b>*STMF</b>	(Default) The code page attribute of the file in the IFS is used.
--------------	---

**1-65534**

The code page to use.

### **Attachment name in email**

This element allows you to specify the name of the attachment as it should appear in the email.

Options are:

**\*FILE** (Default) The name of the attachment is the same as that of the file being attached.

**Attachment\_name** The name to give the attachment in the email.

### **Example:**

Attaching a PDF file:

**SNDCMNMSG...**

**ATTACH('/home/sales/salesreport.pdf \*ATTACH \*BINARY))**

### **Example:**

Embedding an HTML file:

**SNDCMNMSG...**

**ATTACH('/home/sales/salesreport.htm \*EMBED \*HTML))**

### ***MSGSTMF - Stream file containing message***

The **MSGSTMF** (Stream file containing message) parameter allows you to specify the path name of a stream file from which the text of the message should be taken.

In order to use this parameter, specify **MSG(\*STMF)**.

This could be a text file or a complete HTML document. If a database file member is to be used, the name must be given according to IFS naming conventions (e.g. /QSYS.LIB/library\_name.LIB/file\_name.FILE/member\_name.MBR).

Single values:

**\*NONE** The text of the message is not taken from a stream file. Invalid if **MSG(\*STMF)** was specified.

### **Message stream file path**

Options are:

**path-name** Specify the path name of the file containing the message text.

### Code page

The codepage of the data in the stream file specified on the previous element.

Options are:

**\*STMF** The CCSID attribute of the stream file is used to determine the code page.

**1-65534** Specify the codepage.

### Alternate message stream file path

Allows you to specify a different file that will supply the plain text version of the message.

If a value is specified for both path names on the MSGSTMF parameter, the first path is interpreted as supplying the HTML version of an multipart/alternative message and the second path is interpreted as supplying the plain text version. A multipart/alternative message (with both HTML and plain text parts) will be sent irrespective of the message type specified on the MSG parameter, which is ignored.

Options are:

**\*NONE** No alternative plain text version of the message is supplied. If \*BOTH was specified for message type on the MSG parameter, both parts of the multipart/alternative message will be created from the file specified on the first element of the MSGSTMF parameter. Text is converted to or from HTML as required.

**path-name** Specify the path name of the file containing the plain text version of the message.

### Code page

The codepage of the data in the stream file specified on the previous element.

Options are:

**\*STMF** The CCSID attribute of the stream file is used to determine the code page.

**1-65534** Specify the codepage.

## **PRIORITY - Priority**

Options are:

- \*NORMAL** (Default) The message is sent specifying normal priority. When the message arrives, the client email software will not mark it as high or low priority.
- \*HIGH** High priority. When the message arrives, the client email software will mark it as high priority.
- \*LOW** Low priority. When the message arrives, the client email software will mark it as low priority.

## **METHOD - Method**

The method by which the email is delivered.

The default is the single option:

- \*MSF** Use Mail Server Framework. This method sends emails by building a MIME message in a temporary stream file, which is then passed to MSF for processing by means of the **QtmmSendMail** API. MSF then passes the message to SMTP for delivery.

If you experience problems with this method, you can try an alternative method which communicates directly with the SMTP server of the target.

## **MSF or SMTP**

Options are:

- \*SMTP** Selects the alternative method which communicates directly with SMTP on the target, by-passing MSF and SMTP on the system i.  
  
Refer to the section "[How CoolSpools Email Works](#)" above for details of the different methods you can choose for delivering emails.

## **SMTP port number**

The port number to use when communicating with SMTP on the target platform.

Options are:

- \*DFT** The well known port number for SMTP (25).

**port\_number** Specify the port number to use.

Please note that this option is currently made available on an “is-as” unsupported basis.

### **Timeout (seconds)**

The timeout in seconds to enforce when communicating with SMTP on the target platform.

Options are:

**60** The default timeout is one minute.

**timeout\_seconds** Specify the timeout in seconds.

### **CFMDEL - Confirm delivery**

Options are:

**\*NO** (Default) No confirmation of delivery if requested.

**\*YES** The message is sent with an indication that you have requested that the recipient return confirmation of delivery. When the message is opened, if you have not switched off this feature, the client software will either send a confirmation message back to the sender of the email or ask you whether you wish to send such a confirmation.

If **\*YES** is specified, you may specify up to 32 email address/name pairs to receive the confirmation messages.

The default is the single value **\*SENDER**, which indicates that confirmation of delivery should be notified just to the email address specified as the sender on the FROM parameter.

### **SNDMLTMSG - Send multiple messages?**

Options are:

**\*NO** (Default) If you are sending an email to more than one recipient, a single message will be created with multiple recipients. When the message is received, the reader will be able to see the names of all people to whom the message was sent.

**\*YES** If you are sending an email to more than one recipient, multiple separate messages will be sent, one to each

recipient. When the message is received, the reader will see only their own name and will not be able to see the names of all people to whom the message was sent.

## **SAVE - Save options**

The SAVE (Save options) parameter specifies whether the message should be saved to allow later resending with the **RSNCMNMSG** command.

Note that messages sent with SAVE(\*NO) and LOG(\*YES) specified cannot be resent with RSNCMNMSG but can be retrieved by specifying the email identifier on the EMAILID parameter of SNDCMNMSG.

The default is the single option **\*NO**, which indicates that the message will not be saved.

Alternatively, specify a value for the following parameter elements:

### **Save message to allow resend?**

**\*YES** A copy of the email will be saved in a stream file on the system so that it can be resent, if required.

### **Encrypt information?**

**\*YES** The saved email is encrypted. A password must be entered on the PWD parameter.

**\*NO** The saved email is not encrypted.

### **Compress information?**

**\*YES** The saved email is compressed.

**\*NO** The saved email is not compressed

### **Retain for how many days?**

**\*NOMAX** The saved email is retained indefinitely.

**Nbr\_of\_days** Specify the number of days after which the email will be considered “expired” and become eligible for deletion when the **DLTCMNMSG** command is run with the **DLTSAVMSG(\*YES \*MSG)** option.

### **Save in directory**

**\*CURDIR** The email is saved in the current directory of the job running the command.

<b>*HOMEDIR</b>	The email is saved in the home directory of the user running the command.
<b>*PRDDIR</b>	The email is saved in the product directory (/CoolSpools Email).
<b>di_name</b>	Specify the name of the directory where the email should be saved.

### ***LOG - Logging options***

The LOG (Logging options) parameter specifies whether the message should be logged in the CoolSpools Email log file.

There is a single-value option **\*NO**, which indicates that the message will not be logged.

Alternatively, specify a value for the following parameter elements:

#### **Log message?**

<b>*YES</b>	The message will be logged in the CoolSpools Email log files.
-------------	---

#### **Encrypt information?**

<b><u>*NO</u></b>	The logged information is not encrypted.
<b>*YES</b>	The logged information is encrypted. A password must be entered on the PWD parameter.

### ***PWD - Encryption password***

The PWD (Encryption password) parameter lets you supply a password which is used to encrypt saved emails and logged information when encryption is requested for those items.

Saved emails and logged information is encrypted using the MD5 hash algorithm and the RC4 encryption algorithm with the password as the key.

There is a single-value option **\*NONE**, which indicates that no password will be supplied. This value is invalid if encryption has been requested.

The password is case-sensitive. If you wish to retrieve a message using SNDCMNMSG and the EMAILID parameter, or if you wish to resend a message with RSNCMNMSG, you will need to supply the encryption password.

**N.B. CoolSpools Email does not currently provide support for encrypted messages. This password simply relates to the encryption of information stored in**

the message log and saved messages.

## ***CHARSET – Character set encoding***

Specifies the character set to be used to encode the message body part of the email.

Note that other character sets may be used to encode parts of the message, such as attachments, where necessary or as requested.

### ***Character set encoding***

Options are:

<b>*ENVVAR</b>	CoolSpools uses the value of environment variable CS_EMAIL_DFT_CHARSET, which can be set to any of the values listed below, or to a CCSID number.  If the environment variable does not exist, or is set to an invalid value, *CALC will be assumed.
<b>*CALC</b>	CoolSpools Email calculates an appropriate character set in which to encode the message. If the CCSID of the job is a single-byte, Latin-1 encoding, ISO 5589-1 (Latin 1, CCSID 819) is used, otherwise UTF-8 is used.
<b>*UTF8</b>	UTF-8 (CCSID 1208)
<b>*WINDOWS</b>	Windows (CCSID 1252)
<b>*WINDOWSC</b>	Windows CE (CCSID 1250)
<b>*WINDOWSCYRILLIC</b>	Windows Cyrillic (CCSID 1251)
<b>*WINDOWSGREEK</b>	Windows Greek (CCSID 1253)
<b>*WINDOWSTURKISH</b>	Windows Turkish (CCSID 1254)
<b>*WINDOWSHEBREW</b>	Windows Hebrew (CCSID 1255)
<b>*WINDOWSARABIC</b>	Windows Arabic (CCSID 1256)
<b>*WINDOWSBAL TIC</b>	Windows Baltic (CCSID 1257)
<b>*WINDOWSVIET</b>	Windows Viet (CCSID 1258)

<b>*ISOLATIN1</b>	ISO Latin 1 (CCSID 819)
<b>*ISOLATIN2</b>	ISO Latin 2 (CCSID 912)
<b>*ISOCYRILLIC</b>	ISO Cyrillic (CCSID 915)
<b>*ISOARABIC</b>	ISO Arabic (CCSID 1089)
<b>*ISOGREEK</b>	ISO Greek (CCSID 813)
<b>*ISOHEBREW</b>	ISO Hebrew (CCSID 916)
<b>*ISOTURKISH</b>	ISO Turkish (CCSID 923)

### ***Harmonize encoding of msg parts***

Determines what action CoolSpools Email takes if it detects a situation where the different parts of a multipart email have different encodings, for example where the value specified for the character set above is UTF-8 but an embedded HTML attachment has a CCSID of 1252 (Windows). This situation is not well handled by some popular email client systems (e.g. Windows Mail), which will probably display the embedded HTML assuming the character set of the message part, despite the inline attachment having its own character set declaration.

Options are:

<b>*WARN</b>	A warning message is issued to alert the user to the situation.
<b>*YES</b>	CoolSpools changes the encoding of embedded attachments to match that of the message part.
<b>*NO</b>	No action is taken.

### ***OPTIONS - Miscellaneous options***

Specifies miscellaneous options for the command.

Single values

<b>*NONE</b>	No miscellaneous options are specified.
--------------	---

Other values (up to 100 repetitions):

#### ***Option key***

Options are:

<b>*DIAGNOSTIC</b>	Diagnostic mode
--------------------	-----------------

Run the command in diagnostic mode.

Default: \*NO.

Options: \*YES, \*NO.

This option should only be used if instructed to do so by ariadne.

**<:var\_name:>** Defines a variable called var\_name the value of which is specified on the next element.

The variable can be specified (in the form **<:var\_name:>**) wherever CoolSpools variables are supported on this command, e.g. message text and subject line.

If the message text or subject contains this variable marker, it will be replaced at run time with the value specified on the next element.

### **Option value**

Options are:

**character-value** Specify the option value corresponding to the option key.

### **Example:**

```
SNDCMMSG ...  
SUBJECT('Order number <:ORDER_NUMBER:> confirmed')  
OPTIONS(('<:ORDER_NUMBER:>' '12345'))
```

Sends a message with the subject line "Order number 12345 confirmed". The value 12345 could be supplied by a variable in a CL program.

## **CRTCMMMSG command**

Option 2 from the **CoolSpools Email** menu runs the **CRTCMMMSG** (Create CoolSpools Email Message) command which allows you to send an email message.

The same functionality is available from the SNDCMMMSG (Send CoolSpools Email Message) command and the CoolSpools Email Send Message API (Application Program Interface), which are described in a later section of this document. CRTCMMMSG provides a more user-friendly interface for creating email messages.

There are no command parameters. The user is presented with the following screen.

```
-----
                                CoolSpools Email - Create Email Message                                ARIADNE1

From:  Email addr . . support@ARIADNESOFTWARE.CO.UK
      Name . . . . . ariadne software
To:   1 Email addr . .
      Name . . . . .
      Type . . . . . *PRI
      2 Email addr . .
      Name . . . . .
      Type . . . . . *PRI
Subject . . . . .

More...

----- Message -----

More...

F3=Exit    F5=Refresh  F6=Insert  F8=Select  F10=Send  F11=Screen 2
F12=Cancel F14=Delete line
-----
```

The various areas of the screen are:

- From:** Defines the email address and name of the sender. By default, the current user's email address details are retrieved from the system distribution directory.
- To:** A scrolling area allowing the email address(es) and name(s) of the recipient(s) of the email to be entered, as well as a code indicating whether the recipient is a primary recipient, a cc: recipient or a bcc: recipient.
- Subject:** The subject line of the email

**Message:** A scrolling area allowing the text of the message to be input. Enter the special value \*STMF to indicate that the text of the message will be supplied in a stream file (specified on the second screen).

Available command keys are:

- F3=Exit** Exit the application.
- F5=Refresh** Reset the screen
- F6=Insert** Insert a line in the message text area.
- F8=Select** Select email addresses from an address directory.
- F10=Send** Send the message.
- F11=Screen 2** Show the second screen, which provides further email options and attachment functionality/
- F12=Cancel** Return to the previous screen.
- F14=Delete line** Delete a line in the message text area.

Pressing F11 from this first screen displays the second screen, which looks like this:

```

-----
                          CoolSpools Email - Create Email Message                          ARIADNE1

Message format . . . *BOTH          Priority . . . . . N
Confirm delivery . . N             Send multiple msgs . N
Encryption password .
Log message . . . . . Y           Log encrypted . . . . N
Save for resending . N           Save encrypted . . . N
Save compressed . . . Y           Retention period . . 9999 (days)
Save in directory . . *CURDIR

Message stream file .

----- Attachments -----
 1 File name . . . .
   Method . . . . . *ATTACH Content . . . *FILE Codepage . . . *STMF
 2 File name . . . .
   Method . . . . . *ATTACH Content . . . *FILE Codepage . . . *STMF
                                           More...

F3=Exit   F5=Refresh   F10=Send   F11=Screen 1
-----

```

The various fields are:

<b>Message format:</b>	The format in which the message is sent (*TEXT, *HTML or *BOTH).
<b>Priority:</b>	The message priority: N=Normal, H=High or L=Low.
<b>Confirm delivery:</b>	Whether confirmation of delivery is requested: N=No, Y=Yes.
<b>Send multiple msgs:</b>	When sending to more than one recipient, whether a separate message is sent to each recipient or a single message to a list of recipients: N=No, Y=Yes.
<b>Encryption password:</b>	The encryption password to be used for encrypting logged and saved message information.  <b>N.B. CoolSpools Email does not currently provide support for encrypted messages. This password simply relates to the encryption of information stored in the message log and saved messages.</b>
<b>Log message:</b>	Whether details of the message are recorded in the CoolSpools Email message log. N=No, Y=Yes.
<b>Log encrypted:</b>	Whether the message text is encrypted in the message log. N=No, Y=Yes.
<b>Save for resending:</b>	Whether the entire message is saved in its final form so it can be re-sent at a later time. N=No, Y=Yes.
<b>Save encrypted:</b>	Whether the saved message is saved in an encrypted form. N=No, Y=Yes.
<b>Save compressed:</b>	Whether the saved message is saved in a compressed form. N=No, Y=Yes.
<b>Retention period:</b>	A number of days after which the saved message is eligible for deletion. This value is used by DLTCMNMSG when you specify DLTSVMSG(*YES *MSG).

**Save in directory:** The path to the directory in which the email message will be saved. \*CURDIR=current directory.

**Message stream file:** The path to a stream file containing the message to be sent. You must enter \*STMF in the message text area of the first screen to use this facility.

**Attachments:** A scrolling area allowing you to enter the path to one or more files to be attached to the email, the method of attachment, the type of file contents and the codepage. See the ATTACH parameter of the SNDCMNMSG command for details.

Available command keys are:

<b>F3=Exit</b>	Exit the application.
<b>F5=Refresh</b>	Reset the screen
<b>F10=Send</b>	Send the message.
<b>F11=Screen 1</b>	Show the first screen again.
<b>F12=Cancel</b>	Return to the previous screen.

## **RSNCMNMSG command**

Option 4 from the CoolSpools Email menu (Resend email message) prompts the **RSNCMNMSG** (Resend CoolSpools Email Message) command which allows you to resend a previously saved email message.

The difference between CoolSpools Email menu option 3 (Retrieve email message) and 4 (Resend email message) is that option 3 uses a previously logged email message as the basis for creating a new email message.

If you use option 3 (equivalent to SNDCMNMSG with an email identifier specified on the EMAILID parameter), before sending the new message, you can change details (the message text, recipients etc.); the message will be sent with a new email identifier; and any files specified as attachments will be re-imported before the message is created, so that the current version of those attachments will be sent.

On the other hand, using option 4 (equivalent to the RSNCMNMSG command), the original message is resent exactly as it was saved when first created. The recipient list will be the same, as will the message text. Importantly, any attachments will be sent in the form in which they existed at the time the message was created, not in the current form.

Command parameters are as follows:

### ***EMAILID***

The **EMAILD** (Email id to resend) parameter allows you to specify the identifier of a previously saved email to be resent in its original form.

If the email that is to be resent was encrypted when it was saved, you must supply the encryption password on the second element (“Encryption password”).

## **DLTCMNMSG command**

The **DLTCMNMSG** (Delete CoolSpools Email Messages) command provides a purge facility for saved emails and logged email information.

Command parameters are as follows:

### **DLTSAVMSG**

The **DLTSAVMSG** (Delete Saved Messages) parameter lets you specify whether saved messages should be deleted, and, if so, which messages.

The default is the single value **\*NO** which indicates that saved messages should not be deleted.

Alternatively, specify a value for the following parameter elements:

#### **Delete saved messages**

**\*YES** Emails saved when SNDCMNMSG SAVE(\*YES...) was specified will be considered for deletion.

#### **Delete if older than (days)**

**\*MSG** Emails will be considered for deletion based on the the number of days that was specified on the SAVE parameter of the SNDCMNMSG command when the email was created.

**Nbr\_of\_days** Specify the minimum number of days ago that the message must have been saved for it to be eligible for deletion.

### **DLTLOGDTA**

The **DLTLOGDTA** (Delete Log Entries) parameter lets you specify whether logged information for messages should be deleted, and, if so, for which messages.

The default is the single value **\*NO** which indicates that logged information for messages should not be deleted.

Alternatively, specify a value for the following parameter elements:

#### **Delete message log entries**

**\*YES** Information logged for emails when SNDCMNMSG LOG(\*YES...) was specified will be considered for deletion.

## Delete if older than (days)

<b><u>365</u></b>	Log information over 365 days old will be eligible for deletion.
<b>Nbr_of_days</b>	Specify the minimum number of days ago that the message must have been logged for the information to be eligible for deletion.

## Reorganize physical files

<b><u>*NO</u></b>	CoolSpools Email will not attempt to reorganize the log files.
<b>*YES</b>	CoolSpools Email will attempt to reorganize the log files.

## Working with Email Address Lists

Option 10 from the **CoolSpools Email** menu runs the **WRKADRL** (Work with Address Lists) command, which allows you to create and manage address lists. Email address lists provide a means of grouping email addresses so that you can conveniently send messages to groups of people, e.g. departments, customers who have bought a particular product etc.

Selecting option 10 or entering the **WRKADRL** command displays the address lists that you been defined so far.

Address list information is stored in file CM\_ADRLST. If you wish to ensure that only a restricted number of people can maintain address lists, you should set the authorities on this file appropriately, for example by associating an authorization list with it.

```
-----  
Work with Address Lists  
  
Type options, press Enter.  
  1=Create  2=Change  3=Copy  4=Delete  5=Display  6=Print list entries  
  8=Work with list entries  
  
Opt  Address List          Text 'description'  
  
FINANCE                    Finance Department  
IT                          I.T. Department  
MARKETING                  Marketing Department  
PRODUCTION                 Production Department  
SALES                      Sales Department
```

---

Options that can be input against entries in the list are:

- |                                 |  |
|---------------------------------|--|
| <b>1=Create</b>                 | Select this option to create a new address list. This option is available only on the first line of the display. You can enter the name of the new address list in the space available if desired. This is equivalent to running the <b>CRTADRL</b> (Create Address List) command. |
| <b>2=Change</b>                 | Select this option to change the details of an existing address list. This is equivalent to running the <b>CHGADRL</b> (Change Address List) command.  |
| <b>3=Copy</b>                   | Select this option to copy an existing address list and create a new address list. This is equivalent to running the <b>CPYADRL</b> (Copy Address List) command.   |
| <b>4=Delete</b>                 | Select this option to delete one or more address lists. This is equivalent to running the <b>DLTADRL</b> (Delete Address List) command.  |
| <b>5=Display</b>                | Selection this option to display the details of an address list.   |
| <b>6=Print list entries</b>     | Select this option to print the details of an address list and the entries it contains.  |
| <b>8=Work with list entries</b> | Select this option to work with the entries contained in an address list. This is equivalent to running the <b>WRKADRLE</b> (Work with Address List Entries) command.  |

Available command keys are:

- |                   |                                |
|-------------------|--------------------------------|
| <b>F3=Exit</b>    | Exit the application.          |
| <b>F5=Refresh</b> | Refresh the list.              |
| <b>F12=Cancel</b> | Return to the previous screen. |

**F17=Position**

Display the positioner window to allow the list to be positioned to a specified address list.

## **Creating an address list**

The **CRTADRL** (Create Address List) command allows you to create an address list.

Command parameters are as follows:

### ***ADRL***

The **ADRL** (Address list) parameter allows you to specify the name which will be used to identify the address list. The name must be unique and must conform to the normal rules for IBM System i object names, except that it may be up to 20 characters in length.

### ***TEXT***

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the address list, its contents and purpose.

### **Example:**

Creating an address list to store the email addresses of members of the Sales Department.

```
CRTADRL ADRL(SALES) TEXT('Sales Department')
```

## **Changing an address list**

The **CHGADRL** (Change Address List) command allows you to change an existing address list.

Command parameters are as follows:

### ***ADRL***

The **ADRL** (Address list) identifies the address list you wish to change.

### ***TEXT***

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the address list, its contents and purpose.

### **Example:**

Changing the text description on the Sales Department address list.

**CHGADRL ADRL(SALES) TEXT('Sales Department address list')**

## **Copying an address list**

The **CPYADRL** (Copy Address List) command allows you to copy an existing address list and create a new one.

Command parameters are as follows:

### **FROMADRL**

The **FROMADRL** (From address list) identifies the address list you wish to copy.

### **TOADRL**

The **TOADRL** (To address list) identifies the address list you wish to create.

### **TEXT**

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the new address list, its contents and purpose.

### **CPYADRLE**

The **CPYADRLE** (Copy address list entries?) allows you to indicate whether you want to copy the contents of the existing address list or just its description.

Options are:

- |             |   |
|-------------|---|
| <b>*YES</b> | (Default) Yes. The contents of the address list are copied. |
| <b>*NO</b>  | No. The contents of the address list are not copied.        |

### **Example:**

Copy the Sales Department address list to create a combined Sales and Marketing list.

```
CPYADRL  
FROMADRL(SALES)  
TOADRL(SALES_AND_MARKETING)  
TEXT('Sales and Marketing address list')  
CPYADRLE(*YES)
```

## **Deleting an address list**

The **DLTADRL** (Delete Address List) command allows you to delete an existing address list.

Command parameters are as follows:

### **ADRL**

The **ADRL** (Address list) identifies the address list you wish to delete.

### **Example:**

Deleting the Sales Department address list.

**DLTADRL ADRL(SALES)**

## **Converting a Distribution List to an Address List**

Option 11 from the **CoolSpools Email** menu runs the **CVTDSTADRL** (Convert Distribution List to Address List) command which allows you to convert a SNADS distribution list to an address list for use with **CoolSpools Email**.

If you were formerly a user of Office Vision/400, you may have SNADS distribution lists which are equivalent to **CoolSpools Email** address lists and which you do not wish to have to recreate. This command allows you to automate the conversion of your existing SNADS distribution lists and create **CoolSpools Email** address lists from them.

Command parameters are as follows:

### **DSTL**

The **DSTL**(Distribution List) parameter identifies the distribution list you wish to convert.

### **ADRL**

The **ADRL** (Address List) parameter specifies the name of the new address list your wish to create.

Options are:

<b>*DSTL</b>	(Default) The name of the address list is derived from the name of the distribution list by concatenating the two parts of the distribution list name with an intervening underscore. For examples, a distribution list called 'ARIADNE SALES' will become an address list called ARIADNE_SALES.
<b>'Address list name'</b>	Enter a valid address list name of your own choosing.

### **TEXT**

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the new address list, its contents and purpose.

### **HOSTNAME**

The TCP/IP host name to be used when converting distribution list entries to email addresses.

Options are:

<b>*SYS</b>	(Default) The currently defined local host name as specified by the CHGTCPDMN (Change TCP/IP Domain) command is retrieved and used.
<b>'Host name'</b>	Enter a valid TCP/IP host name of your own choosing.

## **DOMAIN**

The TCP/IP domain name to be used when converting distribution list entries to email addresses.

Options are:

<b>*SYS</b>	(Default) The currently defined local domain name as specified by the CHGTCPDMN (Change TCP/IP Domain) command is retrieved and used.
<b>'Domain name'</b>	Enter a valid TCP/IP domain name of your own choosing.

## **Example:**

Converting the old SNADS Sales Department distribution list to an address list.

```
CVTDSTADRL
  DSTL(ARIADNE SALES)
  ADRL(*DSTL)
  HOSTNAME(*SYS)
  DOMAIN(*SYS)
```

*CoolSpools Email* retrieves the contents of the SNADS distribution list and creates a new address list using the value of the ADRL parameter.

For each entry in the address list, *CoolSpools Email* then looks at the System Distribution Directory for the user concerned and constructs an email address according to the following rules:

1. If the "Preferred Address type" is set to \*SMTP, and an SMTP user id and SMTP domain are defined for the user, these are used and the email address will be in the form: SMTPUserId@SMTPDomain.
2. If the "Preferred Address type" is set to \*USRID, the two parts of the user id are concatenated with an intervening period, and these are added to the specified host and domain names to form the email address, e.g. UsrId1.UsrId2@hostname.domainname.

3. If neither of the above rules can be used, the user profile will be added to the host name and domain name specified to give an email address in the form:  
UserProfile@hostname.domainname.

**CoolSpools Email** also derives the name of the user concerned according to the following rules:

1. If a "Preferred name" is available, this is used.
2. If a "First name" and "Last name" are available, these are used.
3. If a "Given name" and "Surname" are available, these are used.
4. If a "User description" is available, it is used.
5. If the user profile has text associated with it, it is used.
6. Otherwise the name is blank.

## **Working with Address List Entries**

Address list entry information is stored in file CM\_LSTENT. If you wish to ensure that only a restricted number of people can maintain address lists, you should set the authorities on this file appropriately, for example by associating an authorization list with it.

The **WRKADRLE** (Work with Address List Entries) command allows you to add and manage address list entries.

There is a single command parameter

### **ADRL**

The **ADRL** (Address list) identifies the address list whose entries you wish to work with.

Alternatively, use the **WRKADRL** (Work with address lists) command and enter option 8=Work with address list entries against the address list in which you are interested.

```
-----  
                                Work with Address List Entries  
                                System:    YOURSYS  
Address list . . . . . :    SALES  
Text 'description' . . :    Sales Department  
Type options, press Enter.  
  1=Add   2=Change   4=Remove   5=Display   6=Print  
Opt      Email address  
  
      Jane.Doe@ariadnesoftware.co.uk  
      Joe.Bloggs@ariadnesoftware.co.uk  
      John.Smith@ariadnesoftware.co.uk  
  
Bottom  
  F3=Exit   F5=Refresh   F12=Cancel   F17=Position to  
-----
```

Options that can be input against entries in the list are:

<b>1=Add</b>	Select this option to add a new address list entry. This option is available only on the first line of the display. You can enter the new email address in the space available if desired. This is equivalent to running the <b>ADDADRLE</b> (Add Address List Entry) command.
<b>2=Change</b>	Select this option to change an existing address list entry. This is equivalent to running the <b>CHGADRLE</b> (Change Address List Entry) command.
<b>4=Remove</b>	Select this option to remove one or more address list entries. This is equivalent to running the <b>RMVADRLE</b> (Remove Address List Entry) command.
<b>5=Display</b>	Selection this option to display the details of an address list entry.
<b>6=Print</b>	Select this option to print an address list entry.

Available command keys are:

<b>F3=Exit</b>	Exit the application.
<b>F5=Refresh</b>	Refresh the list.
<b>F12=Cancel</b>	Return to the previous screen.
<b>F17=Position</b>	Display the positioner window to allow the list to be positioned to a specified address list.

## **Adding an address list entry**

The **ADDADRLE** (Add Address List Entry) command allows you to add an entry to an address list.

Command parameters are as follows:

### **ADRL**

The **ADRL** (Address list) parameter allows you to specify the name of the address list to which the entry is to be added.

### **EMAIL**

The **EMAIL** (Email address) parameter allows you to specify the email address to be added to the list. This must be a valid (i.e. well formed) email address and must be unique within the address list.

### **NAME**

The **NAME** (Name) parameter enables you to enter the name of the person or organization the email address relates to.

### **TYPE**

The **TYPE** (Type) parameter defines the manner in which messages are sent to the email address being added to the list when the address list is used.

Options are:

- |             |   |
|-------------|---|
| <b>*PRI</b> | (Default) Primary recipient.  |
| <b>*CC</b>  | Carbon copy recipient. An *CC recipient receives a copy of the message, and is identified to the primary recipient, but is not the primary recipient. |
| <b>*BCC</b> | Blind carbon copy recipient. An *BCC recipient receives a copy of the message, but is not identified to the primary recipient or *CC recipients.      |

### **Example:**

Adding an entry to the Sales Department address list.

**ADDADRLE**

**ADRL(SALES)**

**EMAIL('Joe.Bloggs@ariadnesoftware.co.uk')**

**NAME('Joe Bloggs')**

**TYPE(\*PRI)**

## **Changing an address list entry**

The **CHGADRLE** (Change Address List Entry) command allows you to change an address list entry.

Command parameters are as follows:

### **ADRL**

The **ADRL** (Address list) parameter identifies the address list in which the entry you wish to change can be found.

### **EMAIL**

The **EMAIL** (Email address) parameter identifies the address list entry you wish to change.

### **NAME**

The **NAME** (Name) parameter enables you to enter the name of the person or organization the email address relates to.

### **TYPE**

The **TYPE** (Type) parameter defines the manner in which messages are sent to the email address being added to the list when the address list is used.

Options are:

<b>*PRI</b>	(Default) Primary recipient.
<b>*CC</b>	Carbon copy recipient. An *CC recipient receives a copy of the message, and is identified to the primary recipient, but is not the primary recipient.
<b>*BCC</b>	Blind carbon copy recipient. An *BCC recipient receives a copy of the message, but is not identified to the primary recipient or *CC recipients.

### **Example:**

Changing an entry in the Sales Department address list.

#### **CHGADRLE**

```
ADRL(SALES)  
EMAIL('Joe.Bloggs@ariadnesoftware.co.uk')  
NAME('Joseph Bloggs') TYPE(*CC)
```

## **Copying address list entries**

The **CPYADRLE** (Copy Address List Entries) command allows you to copy the contents of an existing address list to another address list.

Command parameters are as follows:

### ***FROMADRLE***

The **FROMADRLE** (From address list) identifies the address list whose contents you wish to copy.

### ***TOADRLE***

The **TOADRLE** (To address list) identifies the address list to which you wish to copy the address list entries.

### **Example:**

Copy the contents of the Sales Department address list to a new, combined Sales and Marketing list.

```
CPYADRLE  
  FROMADRLE(SALES)  
  TOADRLE(SALES_AND_MARKETING)
```

## **Removing an address list entry**

The **RMVADRLE** (Remove Address List Entry) command allows you to remove an entry from an address list.

Command parameters are as follows:

### ***ADRLE***

The **ADRLE** (Address list) parameter identifies the address list in which the entry you wish to remove can be found.

### ***EMAIL***

The **EMAIL** (Email address) parameter identifies the address list entry you wish to remove.

### **Example:**

Removing Joe Bloggs from the Sales Department address list.

**RMVADRLE**

**ADRL(SALES)**

**EMAIL('Joe.Bloggs@ariadnesoftware.co.uk')**

## Working with Email Address Directories

Option 20 from the **CoolSpools Email** menu runs the **WRKADDRDIR** (Work with Email Address Directories) command, which allows you to create and manage email address directories.

Email address directories provide a means of storing the email addresses of people to whom you might wish to send emails. **CoolSpools Email** allows you to search and select email addresses from email address directories when you specify TO(\*SELECT) on the SNDCMNMSG command.

Address directory information is stored in file CM\_ADDRDIR. If you wish to ensure that only a restricted number of people can maintain address directories, you should set the authorities on this file appropriately, for example by associating an authorization list with it.

-----  
CoolSpools Email - Work with Address Directories

System: ARIADNE1

Type options, press Enter.

1=Create 2=Change 3=Copy 4=Delete 5=Display 6=Print entries  
8=Work with directory entries

Opt	Address directory	Text 'description'
	*SYSTEM	System email address directory
	ACCOUNTS	Accounts Department email address directory
	PETE	Pete's personal directory
	SALES	Sales Department email address directory

Bottom

F3=Exit F5=Refresh F12=Cancel F17=Position to

-----  
Options that can be input against entries in the list are:

### **1=Create**

Select this option to create a new address directory. This option is available only on the first line of the display. You can enter the name of the new address directory in the space available if desired. This is equivalent to running the **CRTADDRDIR** (Create Address Directory)

command.

**2=Change**

Select this option to change the details of an existing address directory. This is equivalent to running the **CHGADDRDIR** (Change Address Directory) command.

**3=Copy**

Select this option to copy an existing address directory and create a new address directory. This is equivalent to running the **CPYADDRDIR** (Copy Address Directory) command.

**4=Delete**

Select this option to delete one or more address directories. This is equivalent to running the **DLTADDRDIR** (Delete Address Directory) command.

**5=Display**

Selection this option to display the details of an address directory.

**6=Print entries**

Select this option to print the details of an address directory and the entries it contains.

**8=Work with entries**

Select this option to work with the entries contained in an address directory. This is equivalent to running the **WRKADDRDIRE** (Work with Address Directory Entries) command.

Available command keys are:

**F3=Exit**

Exit the application.

**F5=Refresh**

Refresh the list.

**F12=Cancel**

Return to the previous screen.

**F17=Position**

Display the positioner window to allow the list to be positioned to a specified address directory.

## **Creating an address directory**

The **CRTADDRDIR** (Create Address Directory) command which allows you to create an address directory.

Command parameters are as follows:

### ***ADDRDIR***

The **ADDRDIR** (Address directory) parameter allows you to specify the name which will be used to identify the address directory. The name must be unique and must conform to the normal rules for IBM System i object names, except that it may be up to 20 characters in length. The special value \*SYSTEM is also allowed and is recommended for the system-wide directory.

### ***TEXT***

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the address directory, its contents and purpose.

### **Example:**

Creating an address directory to store the email addresses used by members of the Sales Department.

```
CRTADDRDIR  
  ADDRDIR(SALES)  
  TEXT('Sales Department directory')
```

## **Changing an address directory**

Option 12 from the **CoolSpools Email** menu runs the **CHGADDRDIR** (Change Address List) command which allows you to change an existing address directory.

Command parameters are as follows:

### **ADDRDIR**

The **ADDRDIR** (Address directory) identifies the address directory you wish to change.

### **TEXT**

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the address directory, its contents and purpose.

### **Example:**

Changing the text description on the Sales Department address directory.

```
CHGADDRDIR  
  ADDRDIR(SALES)  
  TEXT('Sales Department address directory')
```

## **Copying an address directory**

Option 13 from the **CoolSpools Email** menu runs the **CPYADDRDIR** (Copy Address List) command which allows you to copy an existing address directory and create a new one.

Command parameters are as follows:

### **FROMADDRDIR**

The **FROMADDRDIR** (From address directory) identifies the address directory you wish to copy.

### **TOADDRDIR**

The **TOADDRDIR** (To address directory) identifies the address directory you wish to create.

### **TEXT**

The **TEXT** (Text 'Description') parameter enables you to enter up to 50 characters of free-format text to describe and further identify the new address directory, its contents and purpose.

### **CPYADDRDIRE**

The **CPYADDRDIRE** (Copy address directory entries command) allows you to indicate whether you want to copy the contents of the existing address directory or just its description.

Options are:

- |             |  |
|-------------|--|
| <b>*YES</b> | (Default) Yes. The contents of the address directory are copied. |
| <b>*NO</b>  | No. The contents of the address directory are not copied.        |

### **Example:**

Copy the Sales Department address directory to create a combined Sales and Marketing directory.

```
CPYADDRDIR  
FROMADDRDIR(SALES)  
TOADDRDIR(SALES_AND_MARKETING)  
TEXT('Sales and Marketing address directory')  
CPYADDRDIRE(*YES)
```

## **Deleting an address directory**

The **DLTADDRDIR** (Delete Address List) command allows you to delete an existing address directory.

Command parameters are as follows:

### **ADDRDIR**

The **ADDRDIR** (Address directory) identifies the address directory you wish to delete.

### **Example:**

Deleting the Sales Department address directory.

**DLTADDRDIR ADDRDIR(SALES)**

## **Working with Address Directory Entries**

Address Directory entry information is stored in file CM\_DIRENT. If you wish to ensure that only a restricted number of people can maintain address directories, you should set the authorities on this file appropriately, for example by associating an authorization list with it.

The **WRKADDRDIRE** (Work with Address Directory Entries) command allows you to add and manage address directory entries.

There is a single command parameter

### **ADDRDIR**

The **ADDRDIR** (Address Directory) identifies the address directory whose entries you wish to work with.

Alternatively, use the **WRKADDRDIR** (Work with address directories) command and enter option 8=Work with Address directory entries against the address directory in which you are interested.

```
-----  
CoolSpools Email - Work with Address Directory Entries  
System: ARIADNE1  
Address directory . . : ACCOUNTS  
Text 'description' . . : Accounts Department email address directory  
  
Type options, press Enter.  
  1=Add   2=Change  4=Remove  5=Display  
  
Opt      Email address  
  
        joe.bloggs@customer.com  
        john.doe@customer.com  
  
Bottom  
  
F3=Exit   F5=Refresh   F12=Cancel   F17=Position  
-----
```

Options that can be input against entries in the list are:

- |                  |   |
|------------------|---|
| <b>1=Add</b>     | Select this option to add a new address directory entry. This option is available only on the first line of the display. You can enter the new email address in the space available if desired. This is equivalent to running the <b>ADDADDRDIRE</b> (Add Address Directory Entry) command. |
| <b>2=Change</b>  | Select this option to change an existing address directory entry. This is equivalent to running the <b>CHGADDRDIRE</b> (Change Address Directory Entry) command.  |
| <b>4=Remove</b>  | Select this option to remove one or more address directory entries. This is equivalent to running the <b>RMVADDRDIRE</b> (Remove Address Directory Entry) command.  |
| <b>5=Display</b> | Selection this option to display the details of an address directory entry.   |
| <b>6=Print</b>   | Select this option to print an address directory entry.   |

Available command keys are:

- |                     |  |
|---------------------|--|
| <b>F3=Exit</b>      | Exit the application.  |
| <b>F5=Refresh</b>   | Refresh the list.  |
| <b>F12=Cancel</b>   | Return to the previous screen.   |
| <b>F17=Position</b> | Display the positioner window to allow the list to be positioned to a specified address directory entry. |

## **Adding an address directory entry**

The **ADDADDRDIRE** (Add Address Directory Entry) command allows you to add an entry to an address directory.

Command parameters are as follows:

### ***ADDRDIR***

The **ADDRDIR** (Address directory) parameter allows you to specify the name of the address directory to which the entry is to be added.

### ***EMAIL***

The **EMAIL** (Email address) parameter allows you to specify the email address to be added to the directory. This must be a valid (i.e. well formed) email address and must be unique within the address directory.

### ***FIRSTNAME***

The **FIRSTNAME** (First Name) parameter enables you to enter the first name of the person the email address relates to.

### ***LASTNAME***

The **LASTNAME** (Last Name) parameter enables you to enter the last name of the person the email address relates to.

### ***PREFIX***

The **PREFIX** (Prefix) parameter enables you to enter any part of the name of the person the email address relates to that should precede the first name (such as a title, e.g. Mrs., Dr, Professor etc.).

### ***MIDDLENAME***

The **MIDDLENAME** (Middle name or initials) parameter enables you to enter a middle name or set of initials for the person the email address relates to.

### ***SUFFIX***

The **SUFFIX** (Suffix) parameter enables you to enter any part of the name of the person the email address relates to that should follow the last name (e.g. PhD, Esq. etc.).

## **COMPANY**

The **COMPANY** (Company or organization ) parameter enables you to enter the name of the company or organization to which the person the email address relates to belongs.

## **POSITION**

The **POSITION** (Position or job title) parameter enables you to enter the position, job title or other role of the person the email address relates in the company or organization.

## **EMAILNAME**

The **EMAILNAME** (Email Name) parameter defines the name of the person email address being added to the directory relates to as it should appear when an email sent to that person is delivered. The default is **\*GEN**, which means that CoolSpools Email should generate a name from the **FIRSTNAME** and **LASTNAME** parameters.

### **Example:**

Adding an entry to the Sales Department address directory.

```
ADDADDRDIRE  
  ADDRDIR(SALES)  
  EMAIL('Joe.Bloggs@ariadnesoftware.co.uk')  
  FIRSTNAME('Joe')  
  LASTNAME('Bloggs')  
  COMPANY('ariadne software ltd.')  
  POSITION('Programmer')
```

## **Changing an address directory entry**

The **CHGADDRDIRE** (Change Address Directory Entry) command allows you to change an address directory entry.

Command parameters are as follows:

### ***ADDRDIR***

The **ADDRDIR** (Address directory) parameter identifies the address directory in which the entry you wish to change can be found.

### ***EMAIL***

The **EMAIL** (Email address) parameter identifies the address directory entry you wish to change.

See the ADDADDRDIRE command above for details of the use of the other parameters.

## **Copying address directory entries**

The **CPYADDRDIRE** (Copy Address Directory Entries) command allows you to copy the contents of an existing address directory to another address directory.

Command parameters are as follows:

### **FROMADDRDIR**

The **FROMADDRDIR** (From address directory) identifies the address directory whose contents you wish to copy.

### **TOADDRDIR**

The **TOADDRDIR** (To address directory) identifies the address directory to which you wish to copy the address directory entries.

### **Example:**

Copy the contents of the Sales Department address directory to a new, combined Sales and Marketing directory.

```
CPYADDRDIRE  
  FROMADDRDIR(SALES)  
  TOADDRDIR(SALES_AND_MARKETING)
```

## Removing an address directory entry

The RMVADDRDIRE (Remove Address Directory Entry) command allows you to remove an entry from an address directory.

**Command parameters are as follows:**

### ***ADDRDIR***

The **ADDRDIR** (Address directory) parameter identifies the address directory in which the entry you wish to remove can be found.

### ***EMAIL***

The **EMAIL** (Email address) parameter identifies the address directory entry you wish to remove.

### **Example:**

Removing Joe Bloggs from the Sales Department address directory.

```
RMVADDRDIRE  
  ADDRDIR(SALES)  
  EMAIL('Joe.Bloggs@ariadnesoftware.co.uk')
```

## **SNDCMNICAL command**

Option 31 (Send calendar item (iCalendar format)) prompts the **SNDCMNICAL** (Send iCalendar Item) command which allows you to send a calendar item in iCalendar format.

Calendar items sent using SNDCMNICAL can be received by email client software which support the iCalendar format, such as Outlook (but NOT Outlook Express). Outlook will display items sent by SNDCMNICAL as calendar items and will allow the recipient to perform calendar functions against them. You can use this facility to have the IBM System i schedule meetings, events etc..

Command parameters are as follows:

### **SUMMARY**

Specify summary text for the item. This text appears in the subject line of calendar items received by Outlook.

### **STATUS**

The status of the item.

Options are:

<b>*TENTATIVE</b>	Tentative.
<b>*CONFIRMED</b>	Confirmed.
<b>*CANCELLED</b>	Cancelled.

### **START**

The start date and time of the item.

Options for the date are:

<b>*TOMORROW</b>	(Default) Tomorrow's date is automatically supplied.
<b>*TODAY</b>	Today's date is automatically supplied.
<b>Start_date</b>	Specify the start date.

### **DURATION**

The duration of the item, specified in terms of a number of units and a unit code

Options for the unit code are:

<b>*HOUR</b>	(Default) Hours.
<b>*MINUTE</b>	Minutes.
<b>*DAY</b>	Days
<b>*WEEK</b>	Weeks
<b>*MONTH</b>	Months
<b>*YEAR</b>	Years

## ***LOCATION***

The location of the event, meeting etc. specified as free-form text.

## ***ORGANIZER***

The organizer of the meeting.

There are 2 elements to this parameter: **Email address** and **Name**.

### **Email address**

The email address of the sender of the message and organizer of the event .

The default is \*CURRENT, which tells CoolSpools Email to retrieve the email address of the user running the command from the system distribution direction. See the TO parameter of the SNDCMMSG command for further details.

### **Name**

The name of the sender of the message and organizer of the event .

## ***DESCRIPT***

A free-format description of the event.

If the special value DESCRIPT(\*STMF) is specified, the text can be taken from a stream file specified on the DSCSTMF parameter.

### ***DSCSTMF***

Specifies the path to a stream file containing the free-format description of the event. DESCRIPT(\*STMF) must be specified to use this option.

## **ATTENDEES**

The **ATTENDEES** parameter allows you to specify the email addresses to which the email message should be sent and thereby the people to be invited to participate in the event.

There are 5 elements to this parameter.

### **Email address**

The email address of the attendee.

### **Name**

The name of the attendee.

### **Role**

The attendee's role in the event..

Options are:

- \*REQUIRED**            The person is required to attend
- \*OPTIONAL**            The person's attendance is optional.
- \*INFORMATION**      The message is for information only.
- \*CHAIR**                The person will chair the event.

### **Status**

Specifies the participation status for the calendar user specified by the property

Options are:

- \*ACTION**              (Default) Action is required.
- \*ACCEPTED**            The invitation has been accepted.
- \*TENTATIVE**          The invitation has been tentatively accepted.
- \*DELCLINED**          The invitation has been declined
- \*DELEGATED**          The invitation has been delegated

### **RSVP**

Whether a response is requested.

Options are:

- \*YES                      A response is requested.
- \* NO                      No response is requested.

## **ATTACH**

The **ATTACH** (Attachments) parameter allows you to attach or embed up to 32 files to your email message.

There are 4 elements to this parameter

### **File name:**

This is where you enter the full path name of the file you wish to attach. The path name must be entered in IFS format. In other words, documents stored in shared folders must be referenced in the QDLS file system and database file members must be reference in the QSYS.LIB file system.

### **Content type**

The content type tells **CoolSpools Email** and the receiving email software what type of data the attachment represents.

It is important that the content type is specified accurately, as **CoolSpools Email** will process the attachment differently depending on the value you specify here. For example, if you specify the attachment type as \*TEXT or \*HTML, it will be handled using a technique appropriate to textual data, and may be converted from EBCDIC to ASCII, whereas any other content type will cause the attachment to be handled as binary data and sent unaltered.

Moreover, the receiving software may not handle the attachment appropriately if the content type is specified incorrectly. For example, if you send a PDF file attachment, but specify a content type of HTML, the receiving email software may attempt to display the PDF file as HTML, which will cause unwanted results.

Options are:

- \*FILE                      (Default) The content type will be inferred from the extension part of the file name.

You can control which file extensions are identified as which content types by listing file extensions in the second-level text of CoolSpools Database messages CMQ5000-CMQ5004. Each file extension must be from 1-5 characters long, padded with trailing blanks if less than 5 characters long. For example, message

CMQ5001 controls which file extensions identify image files. This message is supplied with the value:

“JPG JPEG GIF BMP PNG TIF TIFF”.

You may change the text of this message, to add or remove extensions, so long as you ensure that each extension is exactly 5 characters long (including any trailing blanks) and remembering that installing PTFs or a new release will overwrite your changes.

Any file extension not listed in one of these messages will be treated as application binary data.

<b>*BINARY</b>	Application binary. This is the default MIME content type used for data types other than those which have distinct content types of their own (image, audio, video etc.)
<b>*TEXT</b>	Identifies the attachment as plain ASCII text.
<b>*HTML</b>	Identifies the attachment as HTML.
<b>*CSV</b>	identifies the attachment as a CSV or other delimited text file.
<b>*PDF</b>	identifies the attachment as a PDF.
<b>*XML</b>	identifies the attachment as XML.
<b>*IMAGE</b>	identifies the attachment as an image.
<b>*VIDEO</b>	identifies the attachment as a video file.
<b>*AUDIO</b>	identifies the attachment as an audio file.

### Code page

This element allows you to identify the code page of the data in the attachment file. This value is used if the content type is indicated as \*TEXT or \*HTML to convert the data to ASCII format so it can be displayed by the client email software.

Options are:

<b>*STMF</b>	(Default) The code page attribute of the file in the IFS is used.
<b>1-65534</b>	The code page to use.

## Attachment name in email

This element allows you to specify the name of the attachment as it should appear in the email.

Options are:

**\*FILE** (Default) The name of the attachment is the same as that of the file being attached.

**Attachment\_name** The name to give the attachment in the email.

## **PRIORITY**

When the **SNDCMNMSG** command is prompted using F4, the **PRIORITY** (Priority) parameter only appears if you then also press F10 to display additional parameters.

Implementation of the feature will vary from one receiving software program to another.

Options are:

- \*NORMAL** (Default) The message is sent specifying normal priority. When the message arrives, the client email software will not mark it as high or low priority.
- \*HIGH** High priority. When the message arrives, the client email software will mark it as high priority.
- \*LOW** Low priority. When the message arrives, the client email software will mark it as low priority.

## **CLASS**

When the **SNDCMNMSG** command is prompted using F4, the **PRIORITY** (Priority) parameter only appears if you then also press F10 to display additional parameters.

The access classification for the calendar event.

Implementation of the feature will vary from one receiving software program to another.

Options are:

- \*PUBLIC** (Default) The item is marked as publicly viewable.
- \*PRIVATE** The item is marked as private.
- \*CONFIDENTIAL** The item is marked as confidential.

## **CATEGORIES**

When the **SNDCMNMSG** command is prompted using F4, the **PRIORITY** (Priority) parameter only appears if you then also press F10 to display additional parameters.

Free format text defining categories to which the item should be assigned. Separate each category with a comma.

Implementation of the feature will vary from one receiving software program to another.

## **SNDCMNVICAL command**

Option 32 (Send calendar item (vCalendar format)) prompts the **SNDCMNIVAL** (Send vCalendar Item) command which allows you to send a calendar item in vCalendar format.

Calendar items sent using SNDCMNVICAL can be received by email client software which supports the vCalendar format, such as Outlook (but NOT Outlook Express). Outlook will display items sent by SNDCMNVICAL as calendar items and will allow the recipient to perform calendar functions against them. You can use this facility to have the IBM System i schedule meetings, events etc.

Please note that we recommend the use of iCalendar items created using SNDCMNICAL rather than vCalendar items created using SNDCMNVICAL with MS Outlook, as iCalendar provides better functionality.

Command parameters are as per SNDCMNVICAL, with the following exceptions:

### **ATTENDEES**

#### **Role**

Options are:

<b>ATTENDEE</b>	Indicates an attendee at the event
<b>*ORGANIZER</b>	Indicates organizer of the event, but not owner
<b>*OWNER</b>	Indicates owner of the event.
<b>*DELEGATE</b>	Indicates a delegate of another attendee.

#### **Status**

Specifies the participation status for the calendar user specified by the property

Options are:

<b>*ACTION</b>	<b>Event requires action by attendee</b>
<b>*SENT</b>	<b>Event was sent out to attendee</b>
<b>*TENTATIVE</b>	<b>Event is tentatively accepted by attendee</b>
<b>*CONFIRMED</b>	<b>Attendee has confirmed their attendance at the event</b>
<b>*DECLINED</b>	<b>Event has been rejected by attendee</b>

**\*DELEGATED**

**Event has been delegated by the attendee to another\***

## **IMPEMLDTA command**

You can use the IMPEMLDTA (Import Email Data) command supplied with CoolSpools Email V6R1M0 to migrate Communiqué user setup data from Communiqué or CoolSpools PLUS. This provides a means of preserving data that you have set up such as email address lists and email address directories, as well as historical information such as message logs.

The data imported includes:

- Email address lists.
- Email address directories.
- Logged messages.
- Next email identifier.

### ***Restrictions***

The files in the from- and to-libraries must not be in use at the time when this command is run.

Command parameters are as follows:

### ***FROMLIB***

Specifies the library from which the data is to be imported.

Options are:

<b>*CMV1R1</b>	The Communiqué V1R1M0 product library (CMNQUEV1R1).
<b>*CPV5R1</b>	The CoolSpools PLUS V5R1M0 product library (COOLSPV5R1).
<b>name</b>	Specify the name of the library where the Communiqué files are held.

### ***REPLACE***

Specifies whether data in the target files will be replaced if they are not empty.

Options are:

<b>*NO</b>	The data is not replaced. The command will fail if the target files already contain data.
------------	---

**\*YES**

Any data in the target files will be overwritten.

## Using the CoolSpools Email Send API CM\_SNDAPIR

As an alternative to using the **SNDCMNMSG** command, you can also call the **CoolSpools Email** API to send an email message. If you want to send an email from a program written in a language such as RPG or COBOL, this may be easier than using **SNDCMNMSG**. In addition, the length of the free-format message text that can be defined when the API is employed is 4,096 bytes as opposed to 512 bytes when **SNDCMNMSG** is used.

When **CoolSpools Email** is installed, a source file call **CM\_SRCFILE** is included in library **COOLSPV6R1** (or **COOLSPV5R2** if you have installed CoolSpools Plus). This contains two members:

1. An ILE RPG copybook called **CM\_SNDAPIR** which contains the definitions you need to call the **CM\_SNDAPIR** API.
2. A program called **EXAMPLE** which gives an example of how to call the API.

There are two possible forms in which parameters can be passed :

### **CoolSpools Email Send Email Message (CM\_SNDAPIR) API**

#### **Parameter List 1**

**(see source file CM\_SRCFILE, member CM\_SNDAPIR, prototype definition CM\_SndMsg)**

Required Parameter Group:

1	Format of sender structure	Input	CHAR(8)
2	Sender	Input	CHAR(*)
3	Number of recipients	Input	BINARY(4)
4	Format of recipient structures	Input	CHAR(8)
5	Recipient structures	Input	Array 1-32 of CHAR(*)
6	Subject	Input	VARCHAR(256)
7	Format of message structure	Input	CHAR(8)
8	Message structure	Input	CHAR(*)

9	Number of attachments	Input	BINARY(4)
10	Format of attachment structure	Input	CHAR(8)
11	Attachment structures	Input	Array 1-32 of CHAR(*)
12	Priority	Input	CHAR(1)
13	Confirm delivery	Input	CHAR(1)
14	Email message identifier	Output	CHAR(30)
15	Error Code	I/O	CHAR(*)
Optional Parameter Group 1:			
16	Send multiple messages	Input	CHAR(1)
Optional Parameter Group 2:			
17	Number of messages created	Output	BINARY(4)
Optional Parameter Group 3:			
18	Size of email id buffer (next parameter)	Input	BINARY(4)
19	Email id buffer	Output	CHAR(*)
Optional Parameter Group 4:			
20	Stream file containing message text	Input	VARCHAR(512)
Optional Parameter Group 5:			
21	Number of email addresses for confirmation messages	Input	BINARY(4)
22	Format of confirmation email addresses	Input	CHAR(8)
23	Email addresses for confirmation messages	Input	CHAR(*)
Optional Parameter Group 6:			
24	Number of reply-to email address structures	Input	BINARY(4)
25	Format of reply-to structures	Input	CHAR(8)
26	Reply-to email address structures	Input	CHAR(*)

Omissible parameter group 7

27	Format of save options structure	Input	CHAR(8)
28	Save options structure	Input	CHAR(*)

Omissible parameter group 8

29	Format of logging options structure	Input	CHAR(8)
30	Logging options structure	Input	CHAR(*)

Omissible parameter group 9

31	Format of password structure	Input	CHAR(8)
32	Password structure	Input	CHAR(*)

## Required Parameter Group

Note that many of the structures shown below include text fields preceded by a 2-byte length. In languages such as ILE RPG which support variable length character fields, these fields can be defined as variable length. For example, the Subject parameter shown as a structure below can be defined in ILE RPG as 50A VARYING.

### Format of Sender Structure

INPUT; CHAR(8)

The format of the Sender parameter.

Currently only one value is permitted: *CM\_F0100*

### Sender Structure

INPUT; CHAR(\*)

A structure identifying the sender of the message.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of sender email address
2	2	CHAR(128)	Sender email address
130	82	BINARY(2)	Length of sender name
134	86	CHAR(128)	Sender name

### Number of recipients

INPUT; BINARY(4)

The number of recipient structures that follow.

### Format of Recipient Structures

INPUT; CHAR(8)

The format of the Recipient structures that follow.

Currently only one value is permitted: *CM\_T0100*

### Recipient Structures

INPUT; ARRAY 1-32 of CHAR(\*)

An array of from 1 to 32 structures identifying recipients of the message.

Each element in the array is a structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of recipient email address
2	2	CHAR(128)	Recipient email address
130	82	BINARY(2)	Length of recipient name
134	86	CHAR(128)	Recipient name or *ADRL
262	106	CHAR(10)	Recipient type: *PRI, *CC, *BCC or *ADRL

### Subject

INPUT; CHAR(\*)

A structure defining the subject of the message

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of subject text
2	2	CHAR(50)	Subject text

### Format of Message Structure

INPUT; CHAR(8)

The format of the Message structure that follows.

Two values are permitted with this parameter definition:

*CM\_M0100 and CM\_M0200.*

### Message Structure

INPUT; CHAR(\*)

A structure defining the message to be sent, if any, in one of the following formats.

Format for CM\_0100

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of message text
2	2	CHAR(4096)	Message text. Specify *STMF to tell CoolSpools Email to take the text of the message from a stream file defined on optional parameter group 4.
4098	1002	CHAR(5)	Message type: *BOTH, *TEXT or *HTML

Format for CM\_0200

Offset		Type	Field
Dec	Hex		
0	0	POINTER	A pointer to the message text
16	10	BINARY(4)	The length of the message text pointed to by the previous element.
20	14	BINARY(4)	The CCSID of the message text
24	18	CHAR(10)	Message type: *BOTH, *TEXT or *HTML

**Number of attachments**

INPUT; BINARY(4)

The number of attachment structures that follow.

**Format of Attachment Structures**

INPUT; CHAR(8)

The format of the Attachment structures that follow.

Two values are permitted: *CM\_A0100* or *CM\_A0200*

**Attachment Structures**

INPUT; ARRAY 1-32 of CHAR(\*)

An array of from 1 to 32 structures identifying attachments to the message.

Each element in the array is a structure in one of the following formats.

Format for CM\_A0100:

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of attachment path name
2	2	CHAR(512)	Attachment path name
514	202	CHAR(7)	Attachment method: *ATTACH or *EMBED
521	209	CHAR(7)	Content type: *FILE, *TEXT, *HTML, *IMAGE, *VIDEO, *AUDIO, *BINARY
528	210	BINARY(4)	Attachment code page: -1 = *STMF

Format for CM\_A0200:

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of attachment path name
2	2	CHAR(512)	Attachment path name
514	202	CHAR(7)	Attachment method: *ATTACH or *EMBED
521	209	CHAR(7)	Content type: *FILE, *TEXT, *HTML, *IMAGE, *VIDEO, *AUDIO, *BINARY
528	210	BINARY(4)	Attachment code page: -1 = *STMF
532	214	BINARY(2)	Length of attachment name
534	216	CHAR(64)	The name to give the attachment in the email. *FILE=The name is the same as that of the file being attached.

**Priority**

INPUT; CHAR(1)

A code indicating the message priority:

*N* Normal priority

*H* High priority

*L* Low priority

### Confirm delivery

INPUT; CHAR(1)

Whether confirmation of delivery is requested:

*Y* Yes

*N* No

### Email message id

OUTPUT; CHAR(30)

The unique message identifier of the message that was sent, if the call to the API was successful.

This value is the key to the *CoolSpools Email* message log file **CM\_MSGLOG**.

### Error code

INPUT-OUTPUT; CHAR(\*)

A standard IBM System i API error structure in the following format.

Offset		Use	Type	Field
Dec	Hex			
0	0	INPUT	BINARY(4)	Bytes provided
4	4	OUTPUT	BINARY(4)	Bytes available
8	8	OUTPUT	CHAR(7)	Exception ID
15	F	OUTPUT	CHAR(1)	Reserved

16	10	OUTPUT	CHAR(*)	Exception data
----	----	--------	---------	----------------

## Optional Parameter Group 1

### Format of Sender Structure

INPUT; CHAR(1)

When sending an email to more than one recipient, whether a single message should be sent to a list of recipients or multiple messages generated, one per recipient.:

*N* (The default). No. A single message is generated with a list of recipients.

*Y* Yes. Multiple messages, one per recipient are generated.

## Optional Parameter Group 2

### Number of messages created

OUTPUT; BINARY(4)

The number of messages created by CoolSpools Email in response to the request.

## Optional Parameter Group 3

### Size of email id buffer

INPUT; BINARY(4)

The size in bytes of the buffer specified on the following parameter.

### Email id buffer

OUTPUT; CHAR(\*)

A buffer into which CoolSpools Email will place an array of CHAR(30) email identifiers in order to provide the caller with a list of the ids of the email messages generated.

## Optional Parameter Group 4

### Name of stream file containing message text.

INPUT; VARCHAR(512)

If the message text specified for the message structure is \*STMF, CoolSpools Email will read the message text from the file specified here. There is no limit on

the length of the data in the file, which could be an ASCII text file, an HTML file, or a database member (specify database files in IFS format i.e. /QSYS.LIB/library\_name.LIB/file\_name.FILE/member\_name.MBR')

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of the stream file path
2	2	CHAR(512)	Stream file path.

## Optional Parameter Group 5

### Number of email addresses for confirmation emails

INPUT; BINARY(4)

The number of confirmation recipient structures that follow.

### Format of Confirmation Recipient Structures

INPUT; CHAR(8)

The format of the Confirmation Recipient structures that follow.

Currently only one value is permitted: *CM\_C0100*

### Confirmation Recipient Structures

INPUT; ARRAY 1-1024 of CHAR(\*)

An array of from 1 to 1-1024 structures identifying email addresses to which the confirmation of delivery messages should be sent, when confirmation of delivery is requested. The default is to send the confirmation of delivery emails to the sender email address.

Each element in the array is a structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of recipient email address
2	2	CHAR(128)	Recipient email address

130	82	BINARY(2)	Length of recipient name
134	86	CHAR(128)	Recipient name or *ADRL

## Optional Parameter Group 6

### Number of reply-to email address structures

INPUT; BINARY(4)

The number of reply-to email address structures that follow.

### Format of reply-to email address structures

INPUT; CHAR(8)

The format of the reply-to email address structures that follow.

Currently only one value is permitted: *CM\_Y0100*

### Reply-to email address structures

INPUT; ARRAY 1-1024 of CHAR(\*)

An array of from 1 to 1-1024 structures identifying email addresses to which replies to the email should be directed. The default is to send the reply to the sender.

Each element in the array is a structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(2)	Length of recipient email address
2	2	CHAR(128)	Recipient email address
130	82	BINARY(2)	Length of recipient name
134	86	CHAR(128)	Recipient name or *ADRL

## Optional Parameter Group 7

### Format of Save Options Structure

INPUT; CHAR(8)

The format of the save option structure that follows.

Currently only one value is permitted: *CM\_V0100*

### Save options Structure

INPUT; CHAR(\*)

A structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Whether to save the email as a stream file so it can be resent at a later time. Y=Yes, N=No.
1	1	CHAR(1)	Save encrypted. Y=Yes, N=No.
2	2	CHAR(1)	Save compressed. Y=Yes, N=No.
3	3	BINARY(4)	Number of days to retain.
7	7	BINARY(2)	Length of directory name
9	9	CHAR(512)	Name of the directory in which the email will be saved.

### Optional Parameter Group 8

#### Format of Logging Options Structure

INPUT; CHAR(8)

The format of the logging options structure that follows.

Currently only one value is permitted: *CM\_L0100*

#### Logging options Structure

INPUT; CHAR(\*)

A structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Whether to log the sending of the email.

			Y=Yes, N=No.
1	1	CHAR(1)	Log message in encrypted form. Y=Yes, N=No.

## Optional Parameter Group 9

### Format of Password Structure

INPUT; CHAR(8)

The format of the password structure that follows.

Currently only one value is permitted: *CM\_P0100*

### Password Structure

INPUT; CHAR(\*)

A structure in the following format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	The password to use when saving the email in encrypted form when logging message details in encrypted form.

## Parameter List 2

(see source file **CM\_SRCFILE**, member **CM\_SNDAPIP**, prototype definition **CM\_SndMsg2**)

This parameter list is identical to Parameter List 1, with the exception that it allows a larger number of recipients and attachments (1024 instead of 32) and some additional items to be defined (e.g. save and logging options).

### Required Parameter Group:

1	Format of sender structure	Input	CHAR(8)
2	Sender	Input	CHAR(*)
3	Number of recipients	Input	BINARY(4)
4	Format of recipient structures	Input	CHAR(8)
5	Array of recipient structures	Input	CHAR(*)
6	Subject	Input	VARCHAR(50)
7	Format of message structure	Input	CHAR(8)
8	Message structure	Input	CHAR(*)
9	Number of attachments	Input	BINARY(4)
10	Format of attachment structure	Input	CHAR(8)
11	Array of attachment structures	Input	CHAR(*)
12	Priority	Input	CHAR(1)
13	Confirm delivery	Input	CHAR(1)
14	Email message identifier	Output	CHAR(30)
15	Error Code	I/O	CHAR(*)

### Optional Parameter Group 1:

16	Send multiple messages	Input	CHAR(1)
----	------------------------	-------	---------

### Optional Parameter Group 2:

17	Number of messages created	Output	BINARY(4)
Optional Parameter Group 3:			
18	Size of email id buffer (next parameter)	Input	BINARY(4)
19	Email id buffer	Output	CHAR(*)
Optional Parameter Group 4:			
20	Stream file containing message text	Input	VARCHAR(512)
Optional Parameter Group 5:			
21	Number of email addresses for confirmation messages	Input	BINARY(4)
22	Format of confirmation email addresses	Input	CHAR(8)
23	Email addresses for confirmation messages	Input	CHAR(*)
Optional Parameter Group 6:			
24	Number of reply-to email address structures	Input	BINARY(4)
25	Format of reply-to structures	Input	CHAR(8)
26	Reply-to email address structures	Input	CHAR(*)
Omissible parameter group 7			
27	Format of save options structure	Input	CHAR(8)
28	Save options structure	Input	CHAR(*)
Omissible parameter group 8			
29	Format of logging options structure	Input	CHAR(8)
30	Logging options structure	Input	CHAR(*)
Omissible parameter group 9			
31	Format of password structure	Input	CHAR(8)
32	Password structure	Input	CHAR(*)

## Required Parameter Group

As above, except:

### Array of Recipient Structures

INPUT; CHAR(\*)

A buffer containing an array of from 1 to 1024 structures identifying recipients of the message.

### Attachment Structures

INPUT; CHAR(\*)

A buffer containing an array of from 1 to 1024 structures identifying attachments to the message.

## **The CoolSpools Email Database**

**CoolSpools Email** includes a number of database tables which you may find useful. These are listed and explained below.

### **Message Log CM\_MSGLOG**

Message Log file CM\_MSGLOG logs each email message that is sent.

Each message is identified by a 30-byte message identifier. This is the message identifier which is displayed in the completion message sent by the **SNDCMNMSG** command (“CoolSpools Email message XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX sent successfully.” and returned by the **CM\_SNDAPIR** API.

Each log record also contains details:

- when the message was sent
- by which job it was sent
- by which user it was sent
- sender email address
- sender name
- subject
- message text
- message type
- priority
- confirm delivery option

### **Message Recipients CM\_MSGRCP**

Message Recipients file CM\_MSGRCP logs recipients for each email message that is sent.

Each record is identified by the email identifier which is also the key to CM\_MSGLOG.

Each record also contains:

- recipient email address

- recipient name
- recipient type

### **Message Attachments CM\_MSGATC**

Message Attachments file CM\_MSGATC logs attachments for each email message that is sent.

Each record is identified by the email identifier which is also the key to CM\_MSGLOG.

Each record also contains:

- attachment file name
- attachment method
- attachment content type
- attachment code page

## Environment variables

**CoolSpools Email** uses a number of environment variables that can be set to control various aspects of its processing, such as setting user-definable defaults for certain parameters.

The concept of environment variables is described and the environment variables used by CoolSpools Email are fully documented in the [CoolSpools Programmer's Guide](#). However, the following are particularly important and deserve to be highlighted here.

Name	Description	Default value	Other possible values
CS_EMAIL_SEND_ERROR_ACTION	What action CoolSpools Spool Converter takes if it is unable to send an email with an attachment, e.g. because no email address is available or an invalid (not well formed) email address is specified.	*STOP Processing stops immediately. If, therefore, a spooled file is being split into multiple output files, and an email error occurs on one output file, any subsequent files will not be processed.	*CONTINUE A warning message is issued but processing continues and any remaining files will be processed if they can be.  *EMAIL:email The email will be sent to the email address specified after the colon instead. Processing then continues.  *ADRL:list The email will be sent to the email address list specified after the colon instead. Processing then continues.
CS_EMAIL_METHOD	The default method by which emails are delivered	*MSF IBM's Mail Server Framework and the IBM SMTP server are used	*SMTP The CoolSpools SMTP server is used. This is now the recommended method.
CS_EMAIL_SAVE_DAYS	When an email is saved, the default retention period (in days) to assign. The email becomes eligible for deletion using the DLTCMNMSG command with the DLTSVMSG(*MSG) option after the specified number of days.	*NOMAX (no limit)	1-9999
CS_EMAIL_SAVE_ENCRYPTED	Whether, when an email is to be saved, it is saved in compressed form.	*NO The email is not encrypted when it is saved/	*YES The email is encrypted using the password specified on the CS_EMAIL_SAVE_PASSWORD variable.
CS_EMAIL_SAVE_OPTION	Whether, by default, emails are saved after being sent to allow them to be resent later using RSNCMNMSG.	*NO The email is not saved	*YES The email is saved
CS_EMAIL_SAVE_PASSWORD	The password to be used when CS_EMAIL_SAVE_ENCRYPTED is *YES	None.	The password to be used. This must be specified in the form of a hex string as returned by the DSPENCPWD command.