TBS Software Inc.

SMTP-Send™ for z/OS

Release 6.0

User Guide

SMTP-Send and OfficePath/SMTP-Send for z/OS

Copyright

Trademarks

Product names may be trademarks or registered trademarks of their respective owners.
OS/390® and z/OS® are registered trademarks of IBM Corporation. Other company, product, and service names may be trademarks or service marks of others.

Edition Information

Table of Contents

Introduction

Overview of SMTP-Send for z/OS

Have you ever wanted to distribute information such as reports and status notifications to your end users directly from z/OS, OS/390 (MVS) batch, TSO and/or CICS functions?

While simple solutions using file-transfer technologies may at first seem attractive, they create new challenges of their own. For example, you could use FTP to transfer data sets to a LAN. The transferred files then need to be addressed and distributed via traditional e-mail. It is possible to create distribution requests to be processed by IBM's SMTP server task, but these require detailed knowledge of a number of Internet protocols.

SMTP-Send for OS/390 and z/OS solves these problems by allowing your MVS batch, CICS and TSO processes to communicate using industry-standard SMTP. Your applications and operations staff can quickly begin mail-enabling your line of business and operations processes from within the 'glass house' without getting lost in arcane protocols such as TCP/IP, SMTP, POP3, IMAP, etc.

Since all of the processing takes place on the MVS platform you don't need to learn about PC or UNIX-based mail systems. Your operations staff can control the processing using tools they're already familiar with.

SMTP-Send is a one-way communications process from MVS to the outside world. This allows you to easily position the MVS SMTP server behind your firewall to preserve system integrity and prevent viruses entering your secure MVS environment.

Highlights

- sends e-mail from z/OS and OS/390 batch, TSO and CICS applications
- uses text control statements to provide a simple interface to arcane Internet protocols
- connects to all SMTP-compliant e-mail systems including Lotus Notes/Domino and Microsoft Exchange
- simple to install, configure and administer
- central source for distribution lists
- central log of all distributions

Benefits

- easily send e-mail from z/OS and OS/390 with no programming
- uses industry-standard SMTP

Other books you may need

SMTP-Send 6.0 Installation & Reference Guide

IBM TCP/IP for MVS Customization and Administration Guide

z/OS or OS/390 Communications Server IP Configuration Guide

Note: Throughout this book you will see sections prefixed with **OP/SS**. These sections apply only to OfficePath/SMTP-Send. If you are using SMTP-Send, then you may ignore them. Where the term SMTP-Send is used in this book, it also applies to OfficePath/ SMTP.

OP/SS Overview of OfficePath/SMTP

Many IBM OfficeVision/MVS (OV/MVS) customers have mail-enabled line-of-business (LOB) applications that use the underlying e-mail network to distribute information such as reports and status notifications from MVS batch, TSO and/or CICS functions to end-users. OV/MVS traditionally uses a SNADS-based e-mail network, either IBM's DISOSS or TBS Software's OfficePath/SNADS.

Today even as OV/MVS customers migrate their e-mail users to LAN-based systems such as Lotus Notes/Domino and Microsoft Exchange their LOB applications continue to run on MVS. However, once all of their OV/MVS users have been migrated, the cost of maintaining a SNADS network just for this purpose becomes a significant cost issue.

While alternate solutions like file-transfer may at first seem attractive, they create new challenges of their own. For example, customers who use OfficePath/Batch-TSO and/or OfficePath/CICS (or their RAPID equivalents) will need to identify all e-mail enabled LOB applications, then change them to use FTP instead of SNADS. Further, FTP only transfers the files to a LAN. The files still need to be addressed and distributed via traditional e-mail.

OfficePath/SMTP-Send (OP/SS) solves these problems by replacing your SNADS e-mail network with industry-standard SMTP. Because it is designed to work with OfficePath or RAPID, no changes are required to your existing mail-enabled LOB applications. Simply replace OfficePath/SNADS or RAPID/Base with OP/SS and map SNADS addresses to SMTP mail addresses.

Additional Highlights

- replaces the SNADS Send services of DISOSS and OfficePath/SNADS
- no changes required to existing e-mail enabled LOB applications
- SNADS to SMTP address mapping using tables or external directory (LDAP)

Additional Benefits

- preserves your investment in OfficePath/Batch, TSO, CICS and RAPID/Batch, TSO, CICS
- replaces SNADS mail infrastructure with industry-standard SMTP
- reduces overall cost of MVS-to-LAN e-mail connectivity through lower software licensing fees and administrative costs

Other books you may need

OfficePath	RAPID
Installation and Administration Guide	Installation and Administration Guide
Reference and Diagnosis	User Guide
Programming Interfaces	Problem Determination Guide

Soft*Switch Application Toolkit compatibility

You can replace the Soft*Switch Application Toolkit using OfficePath/SMTP-Send and the SMTP-Send SSWDA feature. This feature provides a replacement SSWDA program that accepts Application Toolkit input and converts this input into OfficePath/ SMTP-Send statements and automatically calls OfficePath/SMTP-Send.

Sending messages

To send a message using SMTP-Send, you need to supply the envelope information such as message recipients, subject and other e-mail criteria and supply some text for the actual message. In the simplest case, SMTP-Send reads these control statements from one dataset and the message text from another and creates an input stream that is passed to the SMTP server. Messages may be sent from MVS batch, TSO/ ISPF or CICS. Depending upon the configuration options, the envelope information can be logged for message tracking or audit purposes. See 'Logging Distributions' in the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE, for more information.

You can also invoke SMTP-Send directly from a MVS batch program via LINK or ATTACH, specifying the envelope information and message text in buffers described by the OPSBCALL data structure. See CALLING SMTP-SEND DIRECTLY VIA LINK OR ATTACH for more information

You can also use SMTP-Send to automatically send an email message, showing completion status of each preceding step within any job. See NOTIFICATION OF JOB STATUS (OPSNOTFY) for more information.

OP/SS If you are using OfficePath/SMTP-Send to replace OfficePath or RAPID, all of the SMTP-Send control statements described in this chapter are supported. In addition, extensions have been included to support the OfficePath and RAPID control statements. Refer to SENDING MESSAGES WITH OFFICEPATH/SMTP-SEND for additional information.

SMTP-Send input

SMTP-Send uses control statements to specify the message recipients, subject and other e-mail criteria. SMTP-Send reads these control statements and uses the provided information along with the message text to create an input stream that is passed to the SMTP server. The format and content of the control statements are the same for MVS batch and CICS, the only difference being how the control statements are supplied to SMTP-Send. Information on usage with CICS can be found in the USING SMTP-SEND UNDER CICS chapter. The TSO ISPF dialogs build the required control statements based upon input to the panels and the actual control statements used are not visible to the end user. Refer to USING THE TSO/ISPF DIALOGS for information on usage with ISPF.

Simple SMTP-Send jobstream

```
//SAMPLE   JOB ...
//         EXEC PGM=OPSGS3B
//STEPLIB  DD DISP=SHR,DSN=OPS.vers.LOADLIB
//SYSPRINT DD SYSOUT=*
//CONTROL  DD  *
COMMAND     SEND
TO          user@foo.com
FROM        postmaster@foo.com
SUBJECT     This is a sample subject
            Another line of subject
INTYPE      TEXT
ASA         N
//OPSCONFG DD DISP=SHR,DSN=OPS.vers.OPSCONFG
//MESSAGE  DD *
This is the first line of the report
```

This sample JCL shows how easy it is to send a single message. Information about how to distribute the message is provided in the CONTROL dataset. The message content is in the MESSAGE dataset.

Default settings that apply to all requests are specified in the SMTP-Send configuration dataset. See "Configuring SMTP-Send" in the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE for a description of the configuration settings.

Message text

The message text dataset may be any sequential file or member of a PDS or PDS/E. SMTP-Send examines the DCB characteristics of the dataset to determine how to interpret the information. If the DCB indicates that the dataset contains ANSI-defined print controls (also known as ASA characters), SMTP-Send will format the output text according to the print controls. You can override this default with the ASA control statement. Under CICS, the message text is supplied in a CICS TSQ which has no provision for the indication of the presence of ASA characters.

In addition to the MESSAGE DD statement, you can add up to 4 lines (up to 64 characters each), by using the MESSAGE control statement. These lines will appear at the top of the email message text. This provides a way of customizing the message from within the SMTP-Send batch job.

HTML in message text

Since message text is normally displayed in a mail client, and most mail clients are HTML aware, they will interpret HTML characters in order to format the message for display. In some cases, this is not desirable, as a message may be using these characters without any HTML intention. SMTP-Send can construct the message text so as to prevent these HTML characters from being interpreted as HTML. Configuration and control statements are provided to give users control over how these characters are interpreted.

The default setting can be established by specifying the SMTP-Send configuration statement, "MAPHTMLINTEXT [YES | NO]". The YES value causes SMTP-Send to map the HTML characters into HTML symbolics, which instruct the mail client to display the actual character. The NO value prevents this mapping, thereby allowing the mail client to interpret these characters as HTML.

To override this default setting for any particular SMTP-Send job, the OUTTYPE Control statement parameters, MAPHTML or NOMAPHTML can be specified. These parameters correspond to the configuration statement described above.

Specifying a recipient for replies & non-delivery notices

There are a number of parameters to instruct SMTP-Send where to send replies to your message and non-delivery notices.

If a recipient of your email sends a reply, it will be directed to either of these destinations,

   a) *REPLY-TO* address, if one was specified, or
   b) *FROM* address, or
   c) *ERROR_MSG_TO_NAME* , as specified in the configuration file

If your email is not deliverable to one or more recipients, a 'non-delivery notice' can be sent to a specified location. There are 2 forms of 'non-delivery notice', the original text notice, and the new non-delivery receipt (NDR), which allows the original mail to be resent. The new NDR form is used when the configuration file specifies SEND_NDR=YES.

Either type of non-delivery notice type will be sent to the same destination as are *replies*, described above, but there are two additional destinations which you can use to direct non-delivery notices. This is to facilitate sending non-delivery notices to a different destination than replies.
NDRs can be directed to the *NDR-TO* address, and either form of non-delivery notice can be sent to the *SENDER* address.

   a) *NDR-TO* address (this applies only to the new NDR notice.)
   b) *SENDER* address (used to send any 'non-delivery notice' to a different address than replies)
   c) *REPLY-TO* address, if one was specified, or
   d) *FROM* address, or
   e) *ERROR_MSG_TO_NAME* , as specified in the configuration file

Control Statements

Control statement syntax

-Each control statement consists of a command followed by one or more operands.
-The command must begin in the first column and must be in upper-case.
-The operands must begin in column 11, and are case-insensitive.
-Each operand, if there is more than one, is separated by one or more blanks.

-A line that begins with an asterisk (*) is a comment and is not processed. Likewise, blank lines are ignored.
-Line numbers are ignored if the first non-comment control statement contains line numbers.

-Where multiple choices exist for an operand…
-a vertical bar separates each choice of an operand.
-Required operands are enclosed within curly braces { }. One the choices must be selected.
For example,
A command with a required operand. One of the values must be specified.
ASA { Y | N }

-Optional operands are enclosed within square brackets [ ].
For example,
A command with an optional operand. Neither of the values need be specified.
ATTACH [ASA | NOASA]

-Uppercase letters are used to indicate the minimum characters required to recognize the operand value.

-Values given in *italics* are to be substituted with an appropriate value, either a number, or a value listed in the operand description.
For example,
A command with a required operand, and an optional operand. The actual values for *foreground* & *background* are listed in the description for the command.
BODYCOLOR { *foreground* [,*background*] }

-Operands that contain multiple sub parameters are enclosed within parentheses ( ). If you wish to specify any of the sub parameters, you must enclose them within parentheses.
For example, the ATTACH statement may contain the PDF operand, which can itself contain several sub parameters. These sub parameters may have 1 or more spaces between them.
Each line must contain balanced parentheses, and/or double quotes.
If all the sub parameters you want to specify do not fit on a single line, just repeat the operand on the next line, with the additional sub parameters specified.
Double quotes are required for the ATTACH statement, newname value, when it contains special characters, such as a space. This is described with the syntax for ATTACH.
For example,

Valid sample of syntax illustrating…
-each line has balanced double quotes & parentheses
-PDF operand continued on next line
-flexible spacing among operands, and sub parameters (of PDF in this case)

ATTACH   DDN:MyDD "Attachment name" PDF(Pitch(9) Orient(L))
             PDF(tm(.5) bm(.5) lm(1) rm(1))

-For statements that span more than one line, such as SUBJECT, the command does not need to be repeated on each line. This also applies to statements that are commonly repeated, such as TO, CC & BCC. The following examples are valid

```
Example1:  TO      sales
                   fred@foo.com

Example2:  SUBJECT  this is the subject
                    and this is some more subject.
```

How to specify Control Statements

Control statements are supplied to SMTP-Send in one of two ways, as indicated by the EXEC parm used.

Card image: The control statements are stored in a sequential data set, DDname CONTROL, one control statement per record. The DDname MESSAGE contains the message text to be distributed. Using this method, only one message may be sent per SMTP-Send execution. For compatibility with previous releases, SMTP-Send recognizes DDname CARDFILE as equivalent to CONTROL and LISTDSN as equivalent to MESSAGE.

Card image input is the default in z/OS Batch and TSO. If you want to explicitly indicate the use of card image, add *PARM='C'* [CONTROL] to the EXEC statement in your SMTP-Send JCL stream.

With the message text: The control statements are stored along with the message text in DDname MESSAGE.

To specify this mode, add **PARM='M'** [MESSAGE] to the EXEC statement in your SMTP-Send JCL stream.

The ability to include both control statements and messages in the same input stream is particularly useful when you want to send multiple documents in a single Batch run. The following describes the sequence and format of statements:

1. A start boundary record to indicate that control statements follows. This record should contain at least 35 low-values or null characters (x'00') unless the value is changed with the BOUNDARY statement.
2. Control statements (one statement per record)
3. An end boundary record to indicate the last control statement. This record should contain at least 35 high-values characters (x'FF') unless the value is changed with the BOUNDARY statement.
4. Text records for a single message.
5. A start boundary record to indicate that control statements follows. This record contains at least 35 low-values or null characters (x'00')
6. ... etc. ...

Distribution Request Type Statements

COMMAND SEND

Purpose: Indicates the start of distribution control statements.

COMMAND MERGE { TEMPLATE | BATCH }

Purpose: Invokes the optional HTML Mail Merge feature.

TEMPLATE   (default) indicates that template field substitution is to be performed.

BATCH   indicates that "mail merge" processing is to occur.

BOUNDARY START [x'00' | text]

BOUNDARY END [x'ff' | text]

BOUNDARY text

Purpose: Defines the statement that delimits control statements from data. You need to provide BOUNDARY delimiters when you send multiple messages in the same run, i.e. a batch of messages.

START   Identifies the text string which indicates that the records which follow are control statements.

END   Identifies the text string which indicates that the records which follow are data.

blank   Indicates that the same text is used for both START and END delimiters. This style is similar to that used by MIME. The START text string is created by inserting two hyphen characters at the start of text. The END text string is created by appending the START text string with two hyphen characters. For example, if text is abc, then the START string is --abc and the END string is --abc--.

Notes:
1. BOUNDARY statements, when included, must appear before the COMMAND statement. Each BOUNDARY statement type may only appear once.
2. Trailing blanks are removed from text unless it is enclosed in single quotes.
3. If you use the default START and END characters of x'00' and x'ff', then in Batch/TSO each control statement must begin with a string of 35 of these characters. In CICS, the TSQ record must consist of a single character.
4. If you use OfficePath/SMTP-Send with OV/Fax, then you must use the default START and END characters x'00' and x'ff'.

Envelope Information Statements

BCC user [@domain name]

Purpose: specifies the e-mail address of an additional recipient. If a domain name is not specified, the default domain name from the configuration file is appended to the user value. BCC recipients are not included in the recipient lists of copies of the message sent to the primary and secondary recipients. To distribute to multiple recipients, specify multiple BCC control statements on multiple input records.

Example: `BCC fred@foo.com`

CC user [@domain name]

Purpose: specifies the e-mail address of the secondary or informational recipient. If a domain name is not specified, the default domain name from the configuration file is appended to the user value. To distribute to multiple secondary recipients, specify multiple CC control statements on multiple input records.

Example: `CC sales`

FROM user [@domain name]

Purpose: Specifies the e-mail address of the sender. If a domain name is not specified, the default domain name from the configuration file will be appended to the user value.

REPLY-TO user [@domain name]

Purpose: Specifies the e-mail address to whom any reply will be sent. If not supplied, the FROM value will be used. If a domain name is not specified, the default domain name from the configuration file will be appended to the user value.

SENDER user [@domain name]

Purpose: Specifies the e-mail address to send any error messages regarding this email. If not supplied, the FROM value will be used. If a domain name is not specified, the default domain name from the configuration file will be appended to the user value. See SPECIFYING A RECIPIENT FOR REPLIES & NON-DELIVERY NOTICES.

NDR-TO user [@domain name]

Purpose: Specifies where to send non-delivery receipts (NDRs). Specify this parameter if you want them to go to a different recipient than the REPLY-TO.
The NDR is sent to the first address found using the search order, NDR-TO, SENDER, REPLY-TO, FROM, default value (configuration file value for ERROR_MSG_TO_NAME). See SPECIFYING A RECIPIENT FOR REPLIES & NON-DELIVERY NOTICES.

SUBJECT text

Purpose: Specifies the subject of the message.

Note: The value of the SUBJECT control statement also participates in variable substitution when the COMMAND statement has a value of MERGE or when symbolic variables are encountered.

TO user [@domain name]

Purpose: specifies the e-mail address of the primary recipient. If a domain name is not specified, the default domain name from the configuration file is appended to the user value. To distribute to multiple primary recipients, specify multiple TO control statements on multiple control statements. The number of recipients is limited only by the amount of available storage.

Example: `TO sales`
          `TO fred@foo.com`

Content Description Statements

ASA {Y | N}

Purpose: Indicates if the message text contains ASA print carriage control characters (ANSI print controls) in the first column position. The value Y causes SMTP-Send to paginate as ASA characterers are encountered.
The default value depends on the DCB characteristics of the message text file. If ASA is not indicated in the DCB, the default is N. If ASA is indicated in the DCB, the default is Y.

CHARS nnn

Purpose: Defines the number of characters to be read from each line of input message text. The text is truncated at the specified number of characters. If no CHARS value is specified the maximum number of characters is determined by the record length of the input file.

Example: `CHARS 132`

INTYPE { TEXT | 1403 | HTML | MIXED | BINARY | TEMPLATE { KEYWORD | XML } }

Purpose: Describes the format of input in the MESSAGE file.
TEXT   (default) The data is fixed-pitch text.
1403   Equivalent to TEXT
HTML   The data is in HTML format. SMTP-Send builds MIME headers.
MIXED   The message includes one or more attachments. The MESSAGE file contains TEXT data. The data format for each of the attachments is specified in the corresponding ATTACH control statement.
BINARY   The data is in binary data format.
TEMPLATE   The data is a template that is used by the optional HTML Mail Merge feature.
KEYWORD (default) indicates the data contains a MIME stream, and that the MERGDATA file contains keyword=value substitutions. Substitution matching is not case sensitive.
XML indicates that the data in both MESSAGE and MERGDATA files is in XML format. The variables are defined as XML general entities in MERGDATA file. XML entity references define where the variable data is replaced. Substitution matching is case sensitive.

Note: DOCTYPE is a synonym of INTYPE.

LINES nnn

Purpose: Defines the number of lines per page to be used when interpreting ASA control characters. The default is 66.
Example: `LINES 60`

MARGIN nnn

Purpose: Defines the number of characters to drop from the left of the input message text. The default is zero.

Example: `MARGIN 3`

MESSAGE *text*

Purpose: Allows up to 256 characters of message text that will appear at the top of the email message. Specify up to 4 lines of 64 characters each. The command need only be on the first line. This allows customization of the message from the batch job.
Example:
```
MESSAGE   Line 1 (lines end at the vertical bar) 4 lines x 64 characters |
          LINE 2222222222222222222222222222222222222222222222222222222--|
          LINE 3333333333333333333333333333333333333333333333333333333--|
          LINE 4444444444444444444444444444444444444444444444444444444--|
```

SendIfOnlyMSG { Yes | No }

Purpose: specifies whether or not the email is sent, when a NoData condition exists, but there is data provided in the MESSAGE control statement. Read about the NoData condition, in SPECIAL

CONDITIONS. The default is to send. Use this statement to override the configuration value (described in the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE).

```
OUTTYPE { ASIS | ATTACHMENT | ZIP | PDF | PDFZIP | HTML | HTMLPRE |XMLMERGE |
     RTF[ ( pointsize[-{L|P}] ) ] | NOMIME }
     [ SYMBOLS | NOSYMBOLS ]
     [ MAPHTML | NOMAPHTML ]
     [ EMPTY_ATT | NOEMPTY_ATT ]
```

Purpose: Specifies how the message (specified via MESSAGE DD only) is to be formatted. OUTTYPE must follow any INTYPE statement specified.

ASIS   (default) Data is sent in the same format that it was read using INTYPE. For INTYPE TEXT and 1403 the data is sent as HTML encapsulated in <PRE> tags to preserve any column alignment. For INTYPE HTML, MIME headers for Version, Content-Type and Boundary are created. For INTYPE MIXED the data is sent in multipart MIME with text and attachments.

ATTACHMENT   For INTYPE 1403, TEXT or HTML messages, the text message is sent as a text attachment instead of instream text. The attachment filename is derived from the MVS DSName or DDname or can be specified using a FILENAME control statement. Files that are sent as attachments using the ATTACH control statement are not affected by OUTTYPE controls.

ZIP   Data is compressed into PKZIP-compatible format. The ZIP file is sent as an attachment. Text data is converted to ASCII, binary data is left asis. Not available under CICS.

PDF   Text data is translated from EBCDIC to ASCII and converted into a PDF file. The PDF file is sent as an attachment. For INTYPE BINARY the data is sent as MIME with a single attachment. Document characteristics may be specified using the statements described in PDF Creation.

PDFZIP   Text data is translated into a PDF file then compressed into PKZIP-compatible format. The ZIP file is sent as an attachment. Not available under CICS.

HTML   Text data is sent with MIME headers that indicate the data is text/html. HTML tags <HTML><BODY> and </BODY></ HTML> are added around the text if the input type did not indicate HTML.

HTMLPRE Text data is sent with MIME headers that indicate the data is text/html. The HTML tags <HTML><BODY><PRE> and </PRE></BODY></HTML> are added around the text if the input type did not indicate HTML.

XMLMERGE   XML entities from MERGDATA are substituted in the text data from MESSAGE.

RTF   For INTYPE 1403 or TEXT messages, the message is sent as a text attachment formatted in Microsoft RTF format. OUTTYPE RTF only affects the current SEND command. To send all text attachments in RTF, specify DEFAULT_ATTACHMENT RTF in the SMTP-Send Configuration file. You can then override this configuration setting for a specific SEND request by specifying OUTTYPE NORTF.
You can optionally specify the default pitch and page orientation using subparameters to RTF. Pointsize indicates the size of size of the font in points, L indicates Landscape orientation and P is Portrait orientation. If not specified, the default pitch is obtained from the PITCH control statement or configuration statement.

NOMIME   For INTYPE 1403 or TEXT messages, the message is sent as plain text with no MIME headers. OUTTYPE NOMIME only affects the current SEND command. To send all text data as non-MIME plain text specify USE_MIME_FOR_1403 N in the SMTP-Send Configuration file. You can then override this configuration setting for a specific SEND request by specifying OUTTYPE MIME. Use this setting when sending to pagers, or cellphones.

SYMBOLS | NOSYMBOLS
By default, symbolic variables (keywords prefixed by &) are recognized and replaced in the message text, subject and filenames. Specify NOSYMBOLS to ignore these symbolic variables. Refer to Symbolic Substitution for more information on symbols.

MAPHTML | NOMAPHTML

> MAPHTML - For INTYPE 1403 or TEXT messages only, any strings of characters that may be interpreted as HTML upon receipt will have the characters '<', '&' and '>' translated into the HTML symbolics, "<", "&amp" and ">" respectively. This will prevent the receiving mail client from interpreting these characters as HTML.
>
> You can also cause this behavior without specifying MAPHTML by surrounding the string in exclamation marks, i.e. `!<data>!` will be sent as the HTML string "`<data>`"
>
> NOMAPHTML – Will not replace the above mentioned HTML characters with their HTML symbolic equivalents.
>
> Use this parm to override the configuration setting "MAPHTMLINTEXT YES"

EMPTY_ATT | NOEMPTY_ATT

> This parameter applies only to messages that are to be sent as an attachment. If the message contains no data, this statement controls whether or not an empty attachment is created. Use this parameter to override the value set in the *Include_Attach_Empty* configuration statement, see the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE, for descriptions of the configuration statements.

Create Attachment Statements

Batch/TSO:

```
ATTACH { DDN:ddname | DSN:dsname | HFS:pathname }
      [newname]
      [ASA|NOASA] | [MCC|NOCC]
      [INCLUDE_EMPTY | EXCLUDE_EMPTY | SKIPSEND_EMPTY]
      [LF2crlf | NOLF]
      [RTB | LTB]
      [RTF [ ( Pltch(n) Orient( L | P) ) ]  |  NORTF ]
      [PDF [ ( Blas() PAper() Pltch(n) Orient( L | P ) ) ] ]
      [PDF [ ( TM() BM() LM( ) RM() ) ] ]
      [TRANSLATE]
      [ZIP]
```

CICS:

```
ATTACH { TSQ:tsqname }
      [newname]
      [ASA|NOASA] | [MCC|NOCC]
      [INCLUDE_EMPTY | EXCLUDE_EMPTY | SKIPSEND_EMPTY]
      [LF2crlf | NOLF]
      [RTB | LTB]
      [RTF [ ( Pltch(n) Orient( L | P) ) ]  |  NORTF ]
      [PDF [ ( Blas() PAper() Pltch(n) Orient( L | P ) ) ] ]
      [PDF [ ( TM() BM() LM( ) RM() ) ] ]
      [TRANSLATE]
```

Purpose:	Specifies a file to attach to the message. If you want to attach several files then use a separate ATTACH statement for each file. Due to space limitations on a single control statement, the ATTACH statement may be continued on multiple records by leaving the command name blank.
DDN	The DD name of the file to be attached.
DSN	The dataset name of the file to be attached.
HFS	The HFS pathname of the file to be attached.
TSQ	The CICS TSQ name of the file to be attached.
newname	The name of the attached file as it appears in the email. The case of this value is preserved. If not specified, then the MVS data set, HFS file or CICS TSQ name is used. If the format of the attachment is to be converted, using either of the ZIP, PDF or RTF parameters, then the appropriate filename extension will be appended to this name, if it is not already present. This statement permits the use of symbolics (see SYMBOLIC SUBSTITUTION for details). Notes: 1. If newname contains blanks, or could be confused with one of the other ATTACH keywords enclose the value in double quotes. 2. If a CICS TSQ name contains unprintable characters, then they are translated into their hexadecimal equivalent. For example, a TSQ name of x'C1C2C3C40012345C' becomes ABCD0012345C.

ASA | NOASA
MCC | NOCC   ASA - print control characters, ANSI or machine code, that are found in the input file are used to format the attachment. If the input file is defined with the ASA or MCC RECFM attribute, ASA is assumed unless NOASA is coded on the ATTACH statement.

NOASA - ignores the printer control specification in the input file's DCB. Code NOASA to send a RECFM A or M file as a binary attachment (i.e. to preserve the original file characteristics, EBCDIC with printer control characters)

RTF | NORTF   The file is sent as a text attachment formatted in Microsoft RTF format. If DEFAULT_ATTACHMENT RTF is specified in the SMTP-Send Configuration file, RTF is the default, otherwise NORTF is the default.
You can optionally specify the pitch and page orientation specifying subparameters to RTF.
Pitch() indicates the size of the font in points. The default pitch is obtained from the PITCH control statement or configuration statement.
Specify the page orientation using the Orient subparm. The default is Landscape.
Example:
RTF(PI(12) O(p))   specifies a 12 point font size., with Portrait orientation

Note: The old syntax (point-orient) previously used in the RTF operand has been deprecated, but is still supported.

TRANSLATE mode is always forced on for RTF conversions.
If the input attachment is already in RTF format, you should not specify the RTF keyword.

RTB | LTB   Trailing blanks are removed from translated input lines. RTB is an abbreviation for `RemoveTrailingBlanks`. LTB is an abbreviation for `LeaveTrailingBlanks` which instructs SMTP-Send to not remove trailing blanks.   These keywords are used to override the `OPSCONFG RemoveTrailingBlanks` and `CONTROL RTB & LTB` settings.

LF2crlf | NOLF2crlf   LineFeed characters encountered in HFS text files are replaced with a CRLF sequence. NOLF prevents this substitution. These keywords are used to override the `OPSCONFG ReplaceLFwCRLF` and `LF2CRLF CONTROL` settings. Non-HFS files are not affected by this keyword.

PDF   Indicates that the attached file is to be converted to an Adobe Portable Data file (PDF). Subparameters of the PDF keyword specify the characteristics of the PDF file. The defaults for the PDF characteristics are obtained in the same hierarchy as the specifications described under [PDF Creation](). If a single ATTACH statement doesn't have enough space for all your subparameters, you can specify multiple PDF keywords on continuations of the ATTACH statement. TRANSLATE mode is always forced on for PDF conversions.

INCLUDE_EMPTY | EXCLUDE_EMPTY | SKIPSEND_EMPTY
    If the attachment dataset is empty, this parameter determines whether or not it is attached to the email, and whether the email is sent. Use this parameter to override the configuration statement `Include_Empty_Attach`, for this particular attachment. (In previous releases, these parameters were called, SEND_EMPTY | NOSEND_EMPTY). SKIPSEND_EMPTY indicates that this attachment must be present or else the email will not be sent. If the attachment is empty, or the dataset is missing, the email is aborted with a return code. (In previous releases, this parameter was called, CRITICAL.)

TRANSLATE   The input file is translated from the EBCDIC code page to ASCII before being sent.

ZIP   Indicates that the attached file is to be ZIP compressed. A basic ZIP compression feature that is compatible with PKZIP/PKUNZIP is included in the SMTP-Send software.

Formatting Control Statements

FILENAME *name*

Purpose: Specify a FILENAME for the message, when it is to be sent as an attachment (see the OUTTYPE control statement). If *name* contains blanks or special characters, enclose it within double quotes. In the absence of this statement, the default value is either the DSName of the MESSAGE file, or the DDName for instream MESSAGE files.
This statement permits the use of symbolics (see SYMBOLIC SUBSTITUTION for details).

IMPORTANCE {LOw | NOrmal | HIgh}

Purpose: Specifies the importance of the message. The uppercase letters indicate the minimum number of characters required for SMTP-Send to recognize the value. If IMPORTANCE is not specified, no importance header will be written with the message. The importance header setting is recognized by mail clients which will flag the message according to their convention.

NOMLINKS {Y|N}

Purpose: Deactivates the creation of HTML links from e-mail addresses and URLs found in the subject or body of text messages. Specifying Y instructs SMTP-Send not to create mail links, a specification of N will allow the creation of links.
The default is derived from the `Create_MailTo_Links` OPSCONFG control statement.

LF2CRLF {Y|N}

Purpose: Specifies or overrides whether LineFeed control characters are converted to CarriageReturn LineFeed control sequences in text HFS files. The default is derived from the `ReplaceLFwCRLF` OPSCONFG control statement.

LTB

Purpose: Overrides the OPSCONFG RemoveTrailingBlanks setting to prevent the removal of trailing blanks from text input files.

PITCH {*value*}

Purpose: Specifies the point size used for RTF attachments when no value is specified in the RTF subparameter. The default is 9.

ORIENT {L | P}

Purpose: Specifies the page orientation, L for Landscape, P for Portrait. Used for RTF attachments when no value is specified in the RTF subparameter. The default is Landscape.

NOTICE *identifier*

Purpose: Specifies text of a notice that is appended to the message. If the message contains only an attachment, the notice text will precede the attachment.
Identifier   a 3 character identifier which matches with an identifier of a NOTICE string defined in the OPSCONFG
data set

RECEIPT {Y | N | *user* [*@domain name*]}

Purpose: Requests that confirmation of delivery be returned. If an e-mail address is supplied, the confirmation is returned to the supplied address. Otherwise the confirmation is returned to the address specified in the Reply-To address.

RTB

Purpose: Overrides the RemoveTrailingBlanks configuration file setting to force the removal of trailing blanks from text input files.

SENSITIVITY {PErsonal | PRivate | CONfidential | COMpany-Confidential}

Purpose: Indicates the sensitivity of the message. Sensitivity text is added to either the top or bottom of

the message depending upon the `Sensitivity_Msg` OPSCONFG statement specification. The uppercase letters indicate the minimum number of characters required for SMTP-Send to recognize the value. The text that is inserted is controlled by the `Sensitivity_Txt` OPSCONFG statement

PErsonal	signifies the message is personal and that the Personal Sensitivity text should be inserted.
PRivate	signifies the message is private and that the Private Sensitivity text should be inserted.
CONfidential	signifies the message is confidential and that the Confidential Sensitivity text should be inserted.
COMpany-confidential	signifies the message is company confidential and that the Company Confidential Sensitivity text should be inserted.

Cascading Style Sheet support

CSSSTYLE {Yes | No}

Purpose: Specifies whether Cascading Style Sheets are used to format HTML text.

BODYCOLOR { *background color* [,*foreground color*] }

Purpose: Specifies or overrides background and foreground colors when Cascading Style Sheets are used to format HTML text. You can specify any of the following colors (minimum chars required, are in upper case)

Aqua	BLAck	BLUe	Fuschia
GRAy	GREen	Lime	Maroon
Navy	Olive	Purple	Red
Silver	Teal	White	Yellow

Default: BODYCOLOR   White,Black

FONTFAMILY {*font* [,font...]}

Purpose: Specifies or overrides the Font Family specified for HTML when Cascading Style Sheets are used to format HTML text.
Example: FONTFAMILY sans-serif

HTMLFONT {*none* |1|2|3|4|5|6|7}

Purpose: Specifies the relative font size of text being converted to HTML preformatted text using the HTML <FONT=value> tag. The default value of *none*, indicates that no HTML <FONT> tags are to be included with the text. HTMLFONT is ignored if the data is not being converted from plain text to HTML

PDF Creation control statements

PDF document characteristics are specified using the following control statements. These statements are used when the message text is to be converted to PDF (ie. OUTTYPE PDF). They also form the default values for any ATTACH statements within the same SEND request. Characteristics for files attached with the ATTACH statement may be overridden using subparameters of the PDF keyword.
System wide defaults may be set using the corresponding PDF statements of the Configuration utility.

PDFBIAS {Screen | Print}

Purpose: Screen causes SMTP-Send to create PDF documents suitable for screen display, and ignores any other PDF statements. Text is rendered in 12pt Courier, with page length set to 66 lines and page width based upon the longest line of data read. Top and bottom margins are set to zero. Left and right margins are used as specified in statements.
Print causes SMTP-Send to use the PDF values described below, or their equivalents on the ATTACH statement.
Default: Screen

PDFBOTM {*inches* | *millimeters*mm}

Purpose: Specifies bottom margin
Default: PDFBOTM 0

PDFHEIGHT {*inches* | *millimeters*mm}

Purpose: Specifies paper height. (On the ATTACH PDF statement use the PAPER parameter instead.)
Default: `PDFHEIGHT 11`

PDFLM {inches | millimetersmm}

Purpose: Specifies left margin
Default: `PDFLM 0`

PDFORIENT {Portrait | Landscape}

Purpose: Defines orientation of the page when printed.
Default: `Portrait`

PDFPAPER {LETter | LEGal | LEDger | A3 | A4}

Purpose: Specifies common page sizes.
PDFPAPER specifies only the size of the paper, while the PDFORIENT parameter specifies the page orientation.
Default: `PDFPAPER LETter`

PAPER {LETter | LEGal | LEDger | A3 | A4 | w [x] h}

Purpose: This paper specification can be used in the PDF parameter of the ATTACH control statement. It specifies common page sizes, or an explicit width x height value. Note that height & width can be separated with an 'x', a blank or a comma, and can be specified in units of inches, or millimeters. In previous releases, the WIDTH and HEIGHT statements could be used in the ATTACH PDF parameter.
PDFPAPER specifies only the size of the paper, while the PDFORIENT parameter specifies the page orientation. You can specify the width using the special value -1, meaning that the program will determine the width automatically, using the longest record read from the dataset. When this value is used, the PDFORIENT value is ignored.
Default: `PDFPAPER LETter`

PDFPITCH {12| *pointsize*}

Purpose: Specifies size of the text font in points.

PDFRM {*inches* | *millimeters*mm}

Purpose: Specifies right margin
Default: `PDFRM 0`

PDFTOPM {*inches* | *millimeters*mm}

Purpose: Specifies top margin
Default: `PDFTOPM 0`

PDFWIDTH {*inches* | *millimeters*mm}

Purpose: Specifies paper width. (On the ATTACH PDF statement use the PAPER parameter instead.)
Default: `PDFWIDTH 8.5`

Notes for PDF control statements:

      leading zeros are required for decimal inch values less than 1.
      i.e `PDFTOPM 0.5` should be specified as opposed to `PDFTOPM .5`
      there is no space between the *millimeters* value and the **mm** indicator.
      i.e Specify `PDFPAPER 210mmx297mm`

Examples:
      1) Specifies the standard page layout for a laser printer. with letter paper stock (8.5 in x 11 in) in landscape orientation, capable of printing standard mainframe reports (132 cols x 66 lines) in 9 point Courier font, with top and bottom margins of 1/2 inch and left and right margins of 1/4 inch.

```
PDFBIAS P
PDFPITCH 9
PDFPAPER Letter
PDFORIENT L
PDFTOPM 0.5
PDFBOTM 0.5
```

```
        PDFLM 0.25
        PDFRM 0.25
```

2) Specifies a page layout for ISO A4 paper stock (210 mm x 297 mm) in portrait mode with 12 point Courier font.

```
        PDFBIAS P
        PDFPITCH 12
        PDFORIENT L
        PDFPAPER A4
```

3) Specifies a page layout equivalent to ISO A4 paper stock (210 mm x 297 mm) in portrait mode with 12 point Courier font.

```
        PDFBIAS P
        PDFPITCH 12
        PDFORIENT L
        PDFWIDTH 210mm
        PDFHEIGHT 297mm
```

Control statement examples

This section shows examples of SMTP-Send control statements that are used to perform typical messaging functions.

1. Simple text message with no attachments

The input type defaults to TEXT since there is no INTYPE statement. A text message is sent to user bar@foo.com using the contents of the
MESSAGE file.

```
//         EXEC PGM=OPSGS3B
//CONTROL DD * COMMAND   SEND
TO        bar@foo.com
SUBJECT   Message with no attachments
FROM      your@name.com
//MESSAGE DD *
First line of message #1
    ...
Last line of message #1
```

2. Message with two attachments

Sends a multipart MIME message that includes the contents of the MESSAGE file and two attachments.

```
//         EXEC PGM=OPSGS3B
//CONTROL DD *
COMMAND    SEND
TO        bar@foo.com
FROM      your@name.com
* Sends the sequential dataset HOST.WEEKLY.REPORT.SEQ, as an attachment
* named REPORT.PDF.ZIP. The attached file is in ZIP'd PDF format.
ATTACH    DSN:HOST.WEEKLY.REPORT.SEQ  report.pdf  ZIP  PDF
*
* Sends the PDS member referenced by DDName ATTACH2 in PDF format
* as the second attachment named HOST.REPORTS.PDS.MEMBER1.PDF
* The default PDF characteristics are overridden.
ATTACH    DDN:ATTACH2
          PDF(BIAS(P) PAPER(LEG))
SUBJECT   Text Message with two PDF attachments
//ATTACH2 DD DSN=HOST.REPORTS.PDS(MEMBER1)
//MESSAGE DD *
First line of message #1
```

```
        ...
Last line of message #1
```

3. Multiple sends in one batch job

Sends two separate messages from a single batch job. The first line of x'00' characters, which indicates the beginning of control statements for the first message, may be omitted. Standard end-of-file indicates the end of the second message and the end of the batch of messages to be sent. Optionally you could indicate this explicitly using a third line of x'00's.

```
//         EXEC PGM=OPSGS3B,PARM=M
//MESSAGE DD *
* Start of control statements for first message (35 x'00' characters)
00000000000000000000000000000000000000000000000000000000000000000000
COMMAND    SEND
TO         bar@foo.com
SUBJECT    First message
FROM       your@name.com
* End of control statements for first message (35 x'FF' characters)
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
First line of message #1
     ...
Last line of message #1
* Start of control statements for second message (35 x'00' characters)
00000000000000000000000000000000000000000000000000000000000000000000
COMMAND    SEND
TO         foo@bar.com
SUBJECT    Second message
FROM       your@name.com
* End of control statements for second message (35 x'FF' characters)
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
First line of message #2
     ...
Last line of message #2
```

4. Multiple sends from CICS

Sends two separate messages from a CICS application. The first x'00' character, which indicates the beginning of control statements for the first message, may be omitted.
A CICS Item Error after the second TSQ item indicates the end of the second message and the end of the batch of messages to be sent. Optionally you could indicate this explicitly using a single x'00' character.
CICS TSQ input

```
* Start of control statements for first message (one x'00' character)
00
COMMAND    SEND
TO         foo@bar.com
SUBJECT    First message
FROM       your@name.com
* End of control statements for first message (one x'FF' character)
FF
First line of message #1
...
Last line of message #1
* Start of control statements for second message (one x'00' character)
00
COMMAND    SEND
```

```
TO         bar@foo.com
SUBJECT    Second message
FROM       your@name.com
* End of control statements for second message (one x'FF' character)
FF
First line of message #2
...
Last line of message #2
```

5. Setting your own start and end boundaries

This example uses explicit START and END BOUNDARY text instead of the default hex characters. End-of-File indicates the end of message data, however, you could indicate this explicitly using another `==start of control==`.

```
//         EXEC PGM=OPSGS3B,PARM=M
//MESSAGE DD *
BOUNDARY  START '==start of control=='
BOUNDARY  END ==end_of_control==
* Start of control statements for first message
==start of control==
COMMAND   SEND
TO         bar@foo.com
SUBJECT    First message
FROM       your@name.com
* End of control statements for first message
==end_of_control==
First line of message #1
...
Last line of message #1
* Start of control statements for second message
==start of control==
COMMAND   SEND
TO         bar@foo.com
SUBJECT    Second message
FROM       your@name.com
* End of control statements for second message
==end_of_control==
First line of message #2
...
Last line of message #2
```

6. Using your own boundary

This example uses a single BOUNDARY statement instead of the default boundaries.

```
//         EXEC PGM=OPSGS3B,PARM=M
//MESSAGE DD *
BOUNDARY  my boundary
* Start of control statements for first message
--my boundary
COMMAND   SEND
TO         bar@foo.com
SUBJECT    First message
FROM       your@name.com
* End of control statements for first message
--my boundary--
First line of message #1
```

```
...
Last line of message #1
* Start of control statements for second message
--my boundary
COMMAND    SEND
TO         bar@foo.com
SUBJECT    Second message
FROM       your@name.com
* End of control statements for second message
--my boundary--
First line of message #2
...
Last line of message #2
```

This is a variation on the previous example that shows how to include trailing space characters in the BOUNDARY specification.

```
//         EXEC PGM=OPSGS3B,PARM=M
//MESSAGE DD *
BOUNDARY 'my boundary has trailing spaces    '
* Start of control statements for first message
--my boundary has trailing spaces
COMMAND    SEND
TO         bar@foo.com
SUBJECT    First message
FROM       your@name.com
* End of control statements for first message
--my boundary has trailing spaces    --
First line of message #1
...
Last line of message #1
* Start of control statements for second message
--my boundary has trailing spaces
COMMAND    SEND
TO         bar@foo.com
SUBJECT    Second message
FROM       your@name.com
* End of control statements for second message
--my boundary has trailing spaces    --
First line of message #2
...
Last line of message #2
```

Symbolic Substitution

Symbols are recognized in filenames, subject text and message text but not inside attachments. When SMTP-Send recognizes a symbol the symbol is replaced with the appropriate value. Symbol substitution is independent of the HTML Mail Merge feature variable substitution. Certain date symbols (where noted) may include an offset (positive or negative) to be used when calculating the displayed date from the current date.

&JOB	Jobname (1-8 characters)
&JOBID	JobID (1-6 characters) i.e JOB12345 or S56789
&SYSID	System ID (1-4 characters) i.e. SYS1
&USERID	Userid (1-8 characters) of current user running SMTP-Send
&DATE[{+|-}offset]	Date (1-80 characters)
	- In Filenames (**yymmdd** format, i.e.December 31,2000=001231).
	- In message text or Subject in **month dd,yyyy** format. i.e. January 1, 2000
&ISODATE[{+|-}-offset]	ISO format date (10 characters) in **yyyy-mm-dd** format
&CDATE[{+|-}-offset]	Century date (8 characters) in **yyyymmdd** format
&JDATE[{+|-}-offset]	Julian date (5 characters) in **yyddd** format. i.e. 02365

&SDATE[{+|-}-offset]	Short date (4 characters) in **mmdd** format
&UDATE[{+|-}-offset]	Short date (6 characters) in **mmddyy** format
&WEEK[{+|-}offset]	Current week of year in **Wnn** format, week starts on Sunday, Week 1 starts Jan 1
&ISOWEEK[{+|-}offset]	Current week of year in **Wnn** format, week starts on Monday, Week 1 contains 1st Thursday in January
&MONTH[{+|-}-offset]	Month (3-20 characters) i.e. January
&DAY[{+|-}-offset]	Day of the Week (3-20) characters) i.e. Monday
&TIME	Current time (6 characters) in **hhmmss** format
&ISOTIME	Current time (8 characters) in **hh:mm:ss** format

This functionality is active by default but may be turned off for a specific send with the NOSYMBOLS keyword of the OUTTYPE Control statement

Examples

1. In this example, the message data contains symbols.

```
//        EXEC PGM=OPSGS3B
//MESSAGE DD *
This message was sent by &USERID on &DATE.
( reference: &JOB(&JOBID) )
...
```

…assuming arbitrary values for this example, the resulting message would be…

```
This message was sent by KC9355 on July 20, 2011.
( reference: SSWKLY(JOB03598) )
```

2. In this example, the message data is sent as an attached text file. The subject and filenames contain symbols.

```
//        EXEC PGM=OPSGS3B
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    &CDATE Current status from &JOB(&JOBID)
FILENAME   Status.&SYSID.&JOB.&JOBID.&DATE.txt
OUTTYPE    ATTACHMENT
...
```

…assuming arbitrary values for this example, the resulting message would be…

```
Subject: 20110720 Current status from SSWKLY(JOB03598)
```

…the message text would be within an attachment called,
```
Status.SYS1.SSWKLY.JOB03598.110720.txt
```

3. In this example, a file is converted to a PDF, and sent as an attachment. The attachment filename contains symbols.

```
//ATT1     DD DISP=SHR,DSN=STATUS.DATA(CURR)
//        EXEC PGM=OPSGS3B
```

```
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    &CDATE Current status from &JOB(&JOBID)
ATTACH     DDN:ATT1 PDF ASA  Status.&JDATE.&JOB.&JOBID.PDF
…
```

…assuming arbitrary values for this example, the resulting message would be…

```
Subject: 20110720 Current status from SSWKLY(JOB03598)
```

```
…the message would contain a PDF attachment called,
Status.11202.SSWKLY.JOB03598.PDF
```

Calling SMTP-Send directly via LINK or ATTACH

Your program can call SMTP-Send directly via LINK or ATTACH system calls to OPSGS3M alias of the OPSGS3B program. Information about the message to be sent as well as the message text itself is passed in a structure, OPSBCALL. This structure has a mapping macro of the same name and can be found in INSTLIB member, OPSBCALL.

Most of the text values passed to SMTP-Send are required to be in C string format. That is a series of characters terminated by a single x'00' character. The term STRING in the OPSBCALL structure is used to denote such a format.

You must convert the address of the OPSBCALL structure to a string and pass it as character string by appending the address string to a @ character. This format will allow the string to be parsed by SMTP-Send's C runtime libraries. The sample Assembler program, MVSDRIVR illustrates this calling method as well as most of the information passing conventions.

 The text to be sent is supplied via a pointer to a buffer. This pointer,stored in OPSMSGT, points to a string of characters. No explicit length is provided, a single x'00' character terminates the string.

Because SMTP-Send uses the MVS TCPIP stack to send mail, the current TCPIP parameter file must be allocated to the jobstep. The C runtime will attempt to locate this file itself through dynamic allocation. To improve performance, it can be allocated prior to SMTP-Send invocation to the SYSTCPD DDname. The name of this dataset can be determined by examining the TCPIP started task and noting the dataset specified in the SYSTCPD DD statement.

Any error messages returned by the receiving SMTP server are stored in the memory pointed to by OPSERRM@. OPSERRM@ must be zero on the first call to SMTP-Send.

The memory is allocated as required using the subpool specified in the OPSSUB# field. It is the caller's responsibility to release this storage via FREEMAIN if no more calls will be made to OPSGS3M. This memory is automatically released and reallocated upon a subsequent call to OPSGS3M when the same data structure is repeatedly passed to SMTP-Send.

If an addressing error is encountered, by default, SMTP-Send will continue processing as long as there was one recipient accepted by the receiving SMTP Server. Setting the OPSSTOPONADDRERR flag will force SMTP-Send to stop at the first addressing error.

Overriding the system configuration file (OPSCONFG)

Your SMTP-Send installer will have created a configuration file (OPSCONFG) that defines the options that control the processing of SMTP-Send, and sets a variety of default values.

You can create your own OPSCONFG VSAM file using the OPSCONFG utility if you wish to override the customization performed at installation time. For example, you may want to use an alternate configuration file, when

testing a new e-mail enabled application, or to support jobs or users that require different defaults

For information on the contents of the OPSCONFG file, and the configuration statements syntax, see the SMTP-Send 6.0 Installation & Reference Guide.

Notification of job status (OPSNOTFY)

The OPSNOTFY program makes it easy to notify a user by e-mail when their z/OS batch job ends. You can control the execution of OPSNOTFY using the standard JCL COND parameter so that the e-mail is sent only when a job fails.

The e-mail message is formatted as a table. Each row represents a step in the job. It shows the Jobstep name, Procstepname (if any), completion status and completion code, and the program name from the EXEC PGM= parameter. The subject of the message contains the Jobname and JobID.

```
From:    NOTIFY@foo.com
Date:    24/05/2004 01:48 PM
To:      user@yourdomain.com
Subject: Completion status for job OPSNOTFY(JOB05282)
----------------------------------------------------------

Stepname Procstep  Completion     Program
BR14                Ended   0000  IEFBR14
ASM      ASM        Abended S013  ASMA90
LINK                Flushed N/A   IEWL
GENER               Ended   0000  IEBGENER
```

To use OPSNOTFY add the following JCL as the last step in your jobstream.

```
//OPSNOTFY EXEC PGM=OPSNOTFY,COND=EVEN
//STEPLIB  DD DSN=OPS.LOADLIB,DISP=SHR
//OPSCONFG  DD DISP=SHR,DSN=OPS.OPSCONFG
//SYSOUT    DD SYSOUT=*
//MESSAGE   DD UNIT=3380,DISP=(,DELETE),
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=0),SPACE=(CYL,(1,1))
```

By default, OPSNOTFY sends the e-mail to the user who submitted the job. The SMTP address is created using that user's UserID and the default domain from the SMTP-Send configuration file.

You can customize the defaults for column headings, subject, sending address and other parameters by updating INSTLIB member OPSNOTFC. You can override these defaults using the following execution parameters:

```
Syntax: PARM='{[t][,T=id][,S=subject][,M={S|T|L}]} | idvalue
```

t - controls type of SMTP call.

Values for t:
S - TBS Software Inc. SMTP-Send product (default)
O - TBS Software Inc. OP/SMTP-Send product PIPGS3B
R - TBS Software Inc. OP/SMTP-Send product RAPIDGS3

Note: If using OP/SMTP-Send programs, PIPGS3B or RAPIDGS3, you must either, specify this parm ('O' or 'R' respectively) or customize the OPSNOTFC member. If you do not, you may encounter Authorization Code errors as OPSNOTFY attempts to use the incorrect program.
     T=id specifies receiving userid.

Values for id:

        None -results in 'TO:user' where user is equal to the submitting userid.
        $JOBNAME$ results in 'TO: jobname'
        value results in 'TO:value' N.B. Value must be a complete user@domain email address

`S=subject` specifies subject of email (overrides defaults)

`M={S|T|L}` specifies type of message

Subject - Subject only, no step completion Table

Table - Subject + step completion Table
L - Subject, step Table and JESMSGLG log file as an Attachment
    (JES2 only, requires SDSF batch access)

`idvalue` - same as id but must be >1 character - use as only parameter to preserve compatibility with
OPSNOTFY EXEC parm syntax from previous releases.

For example:
```
//OPSNOTFY  EXEC PGM=OPSNOTFY,COND=EVEN,PARM='T=foo@bar.com'
```

The e-mail will be sent to the user "foo@bar.com". The message header will show that it was sent by
NOTIFY@yourdomain, where the domain is extracted from the SMTP-Send configuration file. You can change the
default sender's e-mail address, the message subject, or the message table headings using the customization job
OPSNOTFC from INSTLIB.

You can send to a group of users in a distribution list. The distribution list must have the same name as the
Jobname. Specify $JOBNAME$ in the PARM parameter on the EXEC for OPSNOTFY, for example:

```
//OPSNOTFY  EXEC PGM=OPSNOTFY,COND=EVEN,PARM='T=$JOBNAME$'
```

SMTP-Send resolves the distribution list either (a) from the SMTP-Send configuration file by an LDAP lookup of an
external directory or (b) using a custom directory lookup program you provide.

You can customize the conditions under which an e-mail is sent using COND parameters. See COND Parameter, in
the IBM JCL Reference manual for more information.

OPSNOTFY is unable to send an e-mail message if the job is cancelled by the operator or the system with an S122
or S222 abend. Sometimes other Sx22 abends also prevent the message step from executing. See Errors that
Prevent Step Execution, Regardless of COND Specifications in the IBM JCL Reference manual for more
information.

Sending messages with OfficePath/SMTP-Send

OP/SMTP-Send replaces your existing OfficePath or RAPID API interface. In order to do this transparently, the
OP/SS program uses the same load module names, DDnames and EXEC parameters as those used by OfficePath
or RAPID. In addition, OP/SS recognizes the same CARDFILE control statements as OfficePath or RAPID.

The EXEC parameters used in OfficePath or RAPID, have the same meaning with OfficePath/SMTP-Send.

PARM='D' - the document/message is in the LISTDSN DD file,
           and control statements are in the CARDFILE DD file.
PARM='P' - both control statements and document/message are in the LISTDSN DD file.

In order to replace your existing SNADS-based OfficePath or RAPID/DISOSS network with one that uses SMTP,
OP/SS needs to map existing 8.8 SNADS-format user addresses to SMTP format, for example from
JSMITH.ACMECORP to jsmith@acme.com. OP/SS lets you maintain these mappings either in an internal table or
using an external directory.

Your OP/SS installer should have created a configuration file (OPSCONFG) that defines these mappings. Using this
system (or default) configuration file you can continue to run your existing e-mail enabled applications without
change.

Sample OfficePath/SMTP-Send jobstream

```
//SAMPLE  JOB
//* CHOOSE THE APPROPRIATE PROGRAM NAME AND LOAD LIBRARY
```

```
//* FOR YOUR INSTALLATION
//  EXEC PGM=PIPGS3B,
//* EXEC PGM=RAPIDGS3,
//   PARM='D'
//STEPLIB  DD DISP=SHR,DSN=PIPS161.LOAD          <===
//*        DD DISP=SHR,DSN=RAPID343.LOAD         <===
//SYSPRINT DD SYSOUT=*
//CARDFILE DD  *
COMMAND   SEND     DOCUMENT
DCA LEVEL 01403
 ASA      N
 CHARS    72
 LINES    60
 OUTPUT   F
RECIPIENT TOUSER   PIPHOST2
          USER2    LOCATN2
USER NAME FROMUSR  PIPHOST1
MESSAGE   This is the message that will be transmitted to
          the user specified in the recipient field. The
          message can be as long as four lines of 64
          characters.
SUBJECT   This is a sample subject
          Another line of subject
//OPSCONFG  DD DISP=SHR,DSN=OPS.vers.OPSCONFG      <===
//LISTDSN   DD *
This is the first line of the report
```

Note: Although SYSPRINT is the standard output file for the batch utilities, SYSOUT will be used if SYSPRINT is not allocated. Some RAPID installations use SYSOUT instead of SYSPRINT.

When OP/SS runs under CICS, standard output is written to the CICS Transient Data destination CSMT.

OfficePath/SMTP-Send Extensions

In addition to the supported OfficePath statements (see OFFICEPATH CARDFILE CONTROL STATEMENTS), OfficePath/SMTP-Send also supports all of the 'regular' SMTP-Send control statements detailed at the beginning of this chapter as well as the configuration statements described in the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE. This means that if you are prepared to alter your existing jobstreams, you can use native SMTP addresses as described below. Use the examples earlier in this chapter as a reference, but don't forget to use the correct program name (PIPGS3B or RAPIDGS3), and specify the appropriate EXEC parms as described above.

Native SMTP addressing

In addition to SNADS 8.8 format addresses OP/SS allows you to specify SMTP-format (RFC 822) addresses directly. OP/SS does not perform any mapping on SMTP-format addresses, however these addresses will be subject to validation by any validation logic you supply in an optional mapping/validation exit.

You can specify addresses for the FROM, REPLY-TO, TO, CC and BCC statements (CC and BCC statements were not supported in
OfficePath or RAPID.)

These control statements appear in the CARDFILE section of your input stream in place of the USERNAME, RECIPIENT and REPLY statements. You can intermix RECIPIENT and TO, CC and BCC statements. If you specify both USERNAME and FROM, the message will appear to have been sent by whichever control statement is encountered last by OP/SS.

OfficePath and RAPID only recognize the RECIPIENT statement. If you provide multiple TO, CC or BCC statements, then you must include the TO, CC or BCC command on each control statement.

For example, if your OfficePath or RAPID application includes these CARDFILE control statements:

```
        USER NAME TSO
        RECIPIENT USER2     LOCATN2
                  TOUSER    PIPHOST1
                  FROMUSR   PIPHOST1
                  TSO1
        REPLY     Y
         TO USER  TSO2
```

you can replace them with:

```
        FROM      tso@tbssoft.com
        TO        sales@tbs2.com
        TO        anyname@keane.uk.com
        CC        support@keane.uk.com
        BCC       tso1@tbssoft.com
        REPLY-TO  tso2@tbssoft.com
```

IBMMail addressing compatability

If your OfficePath/SNADS or RAPID/DISOSS system was connected to IBMMail, you may have used the IBMMail feature that lets you send messages to Internet users. Using this feature you address messages to Node=IBMMAIL, User=INTERNET, then include an Internet Addressing Block (IAB) in the body of the message to indicate the recipients' actual SMTP mail address, for example:

```
        /INTERNET
        /TO sales@tbssoft.com
        /CC support@keane.uk.com
        /REPORT
        /END
```

When OP/SS encounters a message to Node=IBMMAIL, User=INTERNET it locates the IAB in the message body, then uses the SMTP addresses in the TO and CC statements to send the message. Refer to the IBMMail Internet Addressing Guide for more information on the use of IABs.

Notes:

1. The IAB must be included at the beginning of the message. Any data that precedes the IAB is ignored.
2. IBMMAIL format input cannot be specified in a HFS file.

Soft*Switch Application Toolkit Feature

The input to the OP/SS SoftSwitch Application Toolkit Feature program, SSWDA is identical to the input required by the actual SoftSwitch SSWDA program. Once OP/SS program has been installed into the correct STEPLIB data sets, existing JCL with `PGM=SSWDA` will run unchanged.

Use the SoftSwitch Application Toolkit Feature to preserve your existing jobstreams. When you need to create new email applications, you should execute the PIPGS3B program using the native OP/SS control statements. Using the native OP/SS control statements you will not be restricted to the SNADS 8 by 8 addressing and will be able to make us of the additional features provided by SMTP-Send.

Special Conditions

This section describes special circumstances which are likely not of concern to most users. In some cases however, a job infrastructure requires these conditions to be handled and thus are important in controlling your workflow.

NoData condition

The NoData condition occurs when there is neither message text, nor attachments to send. In this condition, the email is not sent, and a non-zero return code is issued.
The MESSAGE control statement is an extension supported by OfficePath/SMTP-Send (see OFFICEPATH CARDFILE CONTROL STATEMENTS),   but can also be useful in any SMTP-Send job. The data provided by this statement is added to the beginning of the message text of the email. Using this statement, can prevent an otherwise NoData condition. You can choose to ignore this data when determining the NoData condition, by using the *SendIfOnlyMSG* configuration and/or control statement.

The following statements relate to the NoData condition.

Configuration Statements

SendIfOnlyMSG   Use this configuration statement if you want to prevent the email being sent when the only data is from the MESSAGE control statement.

RC8_on_NoData   by default a NoData condition results in a return code of 4. Use this configuration statement to change it to 8.

Control Statements
SendIfOnlyMSG   Use this control statement to override the same configuration statement.

Attach_Empty condition

The Attach_Empty condition occurs when one or more attachment datasets are empty or missing. In this condition, the email is normally sent. The following statements relate to this condition.

Configuration Statements
Include_Attach_Empty
Use this configuration statement to control whether or not empty attachment(s) are attached to the email.

RC_on_any_Attach_Empty
Use this configuration statement to set the program return code when any attachment is empty.

RC_on_all_Attach_Empty
Use this configuration statement to set the program return code when all attachments are empty.

SkipSend_on_all_Attach_Empty
Use this configuration statement to prevent sending the email when all attachments are empty.

Control Statements
Attach statement parameters…Include_Empty | Exclude_Empty | SkipSend_Empty
Use the *Include_Empty* or *Exclude_Empty* parameters to control whether or not that attachment is added to the email when it is empty. Use the *SkipSend_Empty* parameter to prevent the email from being sent, if the attachment is empty. These control statements can be used to override the configuration, for that attachment.

Using Data Formats

SMTP-Send includes numerous facilities to handle data in various formats, including HTML, RTF, PDF and ZIP. Various formatting parameters influence how SMTP-Send interprets the data you are sending (the INTYPE parameter), and how it creates the data that is sent to the recipient(s) (the OUTTYPE parameter). The following sections describe the various parameters that apply to each data format, and how you might use formatting controls within the input data.

Working with Text

Text data is the basic format of most mainframe data. Usually, this is EBCDIC encoded data, which may or may not include ASA or MCC characters for spacing control. It is often referred to as 1403 format. This is the default INTYPE.

SMTP-Send by default, will translate EBCDIC data into ASCII prior to sending it. If you have ASCII text as your data, you can prevent this translation by using the INTYPE BINARY control statement. With this statement, SMTP-Send will not alter the data in any way. This would also be used to mail binary data, or whenever you want the data to be sent 'asis'.

If the text uses ASA or MCC spacing control, it would normally be indicated by the RECFM attribute of the dataset containing the text. In this case, SMTP-Send will automatically apply the ASA or MCC spacing indicated. If necessary you can override the file attributes, by using the ASA YES or ASA NO control statements. These statements apply for either ASA or MCC controls within the file being sent.

You can imbed HTML or RTF controls in the text data, but such data would now be considered as INTYPE HTML or RTF. This is discussed in the *Working with HTML* and *Working with RTF* sections that follow.

Preventing HTML interpretation of text characters

By default, message text is sent as HTML encapsulated in <PRE> tags to preserve any column alignment. This applies only to the the message text, any text that is sent as an attachment, will be plain text.
The characters '<', '&' and '>' each have HTML meanings, and are interpreted as HTML controls by an email client.
If your message data contains these, and they are not intended to be HTML, there are several ways to preserve them as plain characters. See the OUTTYPE control statement, for more details.

1. Use OUTTYPE NOMIME. This results in the message being sent without MIME headers, causing the email client to display the entire message as plain text. This renders the entire message in text only, and also prevents any attachments in the mail.

2. Use OUTTYPE MAPHTML. This will cause the characters '<', '&' and '>' to be translated into the HTML symbolics, "<", "&" and ">" respectively. The receiving mail client will display the original characters, without any HTML interpretation.

3. You can also make this the default behavior by using the configuration statement MapHTMLinText

Working with HTML

SMTP-Send supports HTML in message text, which is rendered by most mail clients. The layout of message text will generally be preserved. You can also embed HTML controls within the message text, to affect their appearance. For example, you can encode an HTML link to a web site, or set colors or other formatting.

Adding HTML to your Text data

You can simply include HTML controls within your message to affect how it appears or behaves, when viewed in an email client. When you do this, you would specify the INTYPE control statement with a value of HTML. This tells SMTP-Send that your input data contains HTML controls. Following examples illustrate some uses of HTML in your message data.
```
<B> This text will be displayed as bold </B>
```

Including benign HTML commands

Benign HTML commands are valid HTML commands, that you want displayed as text, and not interpreted as HTML. If for example, your text message describes how to use HTML, you want the reader to see the HTML commands as text. To mark an HTML command as benign, add exclamation marks immediately before and after the HTML command. For example, `!<data>!` would be sent as "`<data>`". The client HTML viewer will display `<` (less than symbol) as < , and `>` (greater than symbol) as >, without treating them as HTML characters.

HTML Links that span multiple lines of message

If you are adding an HTML link into your message data, you should be aware of the record length (LRECL) of the input file containing your message. You must ensure that the link does not exceed the LRECL, as this will result in the link being incomplete. For example, if your HTML link is 139 characters long, it should be within a MESSAGE file having an LRECL that is greater than 139.

Working with PDF

SMTP-Send can convert your message, or any text file you wish to attach, into a PDF file. These would appear as attachments in the mail client, and can then be viewed using a PDF reader, such as Adobe's Acrobat Reader. These attachments may also be zipped by adding the ZIP parameter to the ATTACH statement.

If you are storing PDF files on the mainframe, these can be mailed, as is, using the INTYPE BINARY control statement. For input datasets that use ASA or MCC controls for spacing, use the ASA parameter to have the spacing interpreted in the PDF file.

Specifying PDF parameters

As is the case with many SMTP-Send parameters, there are 3 levels at which you can specify PDF parameters.

Configuration File
-applies to all SMTP-Send jobs using the same configuration file, establishes your default values.
-contains statements for formatting the message as a PDF attachment.
-see the SMTP-SEND 6.0 INSTALLATION & REFERENCE GUIDE for descriptions

Control Statements
-apply only to the current SMTP-Send job, including the message and all attachments
-overrides configuration file values
-see PDF CREATION CONTROL STATEMENTS for descriptions

Attach Statement
-applies to the current attachment only
-overrides control statement values
-see CREATE ATTACHMENT STATEMENTS for descriptions

For descriptions of all PDF creation statements, see PDF CREATION CONTROL STATEMENTS

Page considerations

For PDFs that will be viewed only on screen, you need only specify *PDF(BIAS(screen))*, and all other attributes will be set (12 point text, automatic page width (set to longest line of data), zeroed top and bottom margins). In this mode, page size becomes meaningless, as the PDF reader can scroll vertically and horizontally.

If you anticipate printing the PDF, then you may want to control the layout, using *PDF(BIAS(print))*. This allows you to specify the other PDF parameters to ensure a good layout. Determine the page size, margin space, point size and orientation to ensure the source data fits on the page as you want it. If your source data contains ASA or MCC spacing controls, they can be implemented in the PDF also.

Text wrapping…
The program calculates the number of characters that can fit on a line, using the selected text point size, and accounting for the margins and page size. Lines which exceed this length are wrapped to the next line. Normally, it is preferable to select your paper, point size & margins to avoid wrapping.
For example, to fit a 132 character wide file, onto a letter pagesize in landscape mode (11" width), with a point size of 9 you could have up to 1" of total margin space. 9 point text provides for 13.33 characters per inch, so 10" would just fit the 132 characters.

Page width…
The page width is specified using the PAPER parameter, where you indicate a standard paper size, or explicit dimensions. You can use automatic page width by specifying -1 as the width (eg PDF(PAPER(-1 x 11)). This causes the program to track the longest line read, and sets the page width, according to the given point size & margins.

Adobe reader can have problems displaying data when the page width approaches 200 inches. This can occur when your source dataset has a very long record length (eg. Lrecl=4096). Use the removetrailingblanks (RTB) parameter to prevent trailing blanks from contributing to the page width.

Some useful metrics…
Horizontal density…characters per inch = 120 / pointsize
Vertical density…lines per inch = 72 / pointsize
Example, text pointsize = 12   characters per inch=10, lines per inch=6
Factoring in your page size and margins, you can ensure a good fit of your source data.

Lines per page…
If you specify the number of lines per page (LINES control statement), this will be compared with the program calculated value. The LINES value will only take effect if it is equal to, or less that the calculated value. Generally there is no need to specify this statement.

Examples

Send a dataset containing ASA controls, as a .pdf attachment. The attachment name will default to *HLQ.DOCS.SAMPLE1.PDF*. The PDF parameters specify a papersize of -1 x 11, meaning the width will be calculated based upon the longest line read from the dataset, and the height is 11". The orientation is specified as Portrait.

```
//         EXEC PGM=OPSGS3B
//ATT1      DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample PDF document
ATTACH     DDN:ATT1 TRANS ASA
           PDF( pa(-1x11) o(p) )
```

Provide your own name for the attachment, *My First Sample.pdf*. In this example, the *letter* papersize is specified with a *landscape* orientation. A left margin of 1" is requested, the other margins will use the configuration values.

```
//         EXEC PGM=OPSGS3B
//ATT1      DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample document.
ATTACH     DDN:ATT1 "My First Sample" TRANS
  PDF( pa(let) o(l) lm(1) )
```

In this example, we specify a the font point size to ensure that 132 characters can be displayed on a landscape oriented letter pagesize. With a pagewidth of 11", and margins of 0.5" on each side, there is 10" of available width to print characters. A font point size (parm pitch) of 9 is required to fit 132 characters in the 10" of width we have. (note that 120/pointsize gives the characters per inch, in this case, 120/9 = 13.33 x 10" = 133.3 chars would fit in our 10" of width)

```
//         EXEC PGM=OPSGS3B
//ATT1      DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample document.
ATTACH     DDN:ATT1 "My First Sample" TRANS ASA
           PDF( pa(let) o(l) pi(9) )
```

Working with RTF

Overview

SMTP-Send can convert your attachment from EBCDIC text to a basic RTF format, using carriage controls to control the spacing. These attachments can be viewed, and edited using Microsoft Word.

Specifying RTF parameters

You can specify the *Pitch* (point size) and *Orient* values to affect the conversion to RTF. These can be defined in your configuration file, and would be the default values unless overridden by the following control statements.

For the message text…
OUTTYPE RTF(pitch – orient)
> format the message text as an RTF attachment, using the given parameters

Filename	the name to be applied to the attachment.
ASA	message text contains ASA controls, which will be used to affect spacing
Pitch	set the point size of the text (if not provided in the RTF() parameter)
Orient	set page orientation (if not provided in the RTF() parameter)

For an attachment…
ATTACH newname RTF(pitch() orient()) ASA|NOASA*
> Format the attachment as RTF, using the specified options
> *newname* allows you to override the default naming for the attachment
> *If the input file is defined with the ASA or MCC RECFM attribute, ASA is
> assumed unless NOASA is coded on the ATTACH statement.

Examples

Send a dataset containing ASA controls, as an .rtf attachment. The attachment name would normally default to *HLQ.DOCS.SAMPLE1.RTF*, but in this example, the given newname will produce an attachment name of *Jobname(Jobid) Docs Sample1.rtf.*This takes advantage of SYMBOLIC SUBSTITUTION to insert the current values for *Jobname* and *Jobid.* The RTF parameters specify a pitch value of 10, and orientation is specified as portrait.

```
//        EXEC PGM=OPSGS3B
//ATT1     DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample RTF document
ATTACH     DDN:ATT1 "&JOB(&JOBID) Docs Sample1" TRANS ASA
           RTF( pi(10) o(p) )
```

Send the message as an .rtf attachment. The attachment name is specified in the *FILENAME* parameter, and will automatically be appended with the .rtf suffix, if it doesn't already have one. The pitch value is 12, and orientation is specified as portrait.

```
//        EXEC PGM=OPSGS3B
//ATT1     DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a the message text as an RTF attachment
FILENAME   My Message
PITCH      12
ORIENT     P
```

Working with ZIP

Overview

SMTP-Send can compress your attachments into a PKZIP compatible format that can be accessed by any Zip/Unzip utility. The zip feature is not available in the CICS environment.

An attachment can optionally be converted into RTF or PDF format, before being compressed into a zip file.
When you specify the ZIP option, SMTP-Send will ensure that the attachment name ends with *.zip*. This ensures that the email recipient can easily view or extract the contents.

The ZIP function will convert filenames within the ZIP file, uppercase characters, and will replace any blanks with underscore characters. This would apply to values you provide in either the FILENAME or ATTACH statements.

Zipping attachments

You can have any attachment compressed, as a ZIP file, simply by adding ZIP to your ATTACH statement.
The filename within the ZIP file will be the name specified in the *newname* value on the ATTACH statement. (See description of the *newname* parameter, in CREATE ATTACHMENT STATEMENTS for default naming values.). This statement permits the use of symbolics (see SYMBOLIC SUBSTITUTION for details).
If the attachment is converted into RTF or PDF format, the filename will end with *.rtf* or *.pdf*, respectively, before being zipped.

Zipping your attached message

If you are sending your message text as a compressed ZIP attachment, specify the OUTTYPE ZIP statement. You can convert your message text into PDF format and then have it zipped, using the OUTTYPE PDFZIP statement. (*Note: the MESSAGE DD statement must reference a sequential file, not a PDS. If you want to ZIP an entire PDS, you must use the ATTACH statement, as described below.*)
The filename within the ZIP file will be the value specified by the FILENAME statement. This statement permits the use of symbolics (see SYMBOLIC SUBSTITUTION for details). For the default filename values, see the description of the ATTACHMENT parameter, of the OUTTYPE statement.

Zipping a PDS

You can attach an entire PDS to your message, as a zip file attachment. You can do this by using the ATTACH statement, specifying either a DDN or DSN value that refers to a PDS (or PDSE). You must specify the ZIP operand, as it is not possible to send a PDS unless it is zipped. If the PDS contains text data that you want converted into ASCII, then use the TRANSLATE operand on the ATTACH statement.
The filename within the ZIP file will be the name specified in the *newname* value on the ATTACH statement. (See description of the *newname* parameter, in CREATE ATTACHMENT STATEMENTS for default naming values.). This statement permits the use of symbolics (see SYMBOLIC SUBSTITUTION for details).

Examples

Send a text attachment as a .zip file. The attachment name is *HLQ.DOCS.SAMPLE1.ZIP*, and it contains the file *HLQ.DOCS.SAMPLE1*.

```
//         EXEC PGM=OPSGS3B
//ATT1     DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample document.
ATTACH     DDN:ATT1 TRANS ZIP
```

Send the same text attachment as a .zip file, but give it a different name. The attachment name is *My First Sample.zip*, and it will contain the file *MY_FIRST_SAMPLE*.

```
//         EXEC PGM=OPSGS3B
//ATT1     DD DISP=SHR,DSN=HLQ.DOCS(SAMPLE1)
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
```

```
TO         betty@acme.com
Subject    Here is a sample document.
ATTACH     DDN:ATT1 "My First Sample" TRANS ZIP
```

Send the message text as an attached zip file. The attachment name is *MESSAGE.ZIP*, and it contains the file *MESSAGE*.

```
//          EXEC PGM=OPSGS3B
//MESSAGE  DD *
  Here is my message text.
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is an attached zipped message.
OUTTYPE    ZIP
```

Send the message text as an attached zipped PDF file, and give it our own name. The message attachment name is *My Message.zip*, and it contains the file *MY_MESSAGE.PDF*. Also attach a dataset as a zipped PDF. This attachment will be *My Attach.zip*, and it contains the file, *MY_ATTACH.PDF*

```
//          EXEC PGM=OPSGS3B
//MESSAGE  DD *
  Here is my message text.
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is an attached zipped message.
OUTTYPE    PDFZIP
FILENAME   "My Message"
ATTACH     DSN:HLQ.DOCS(SAMPLE2) "My Attach" TRANS PDF ZIP
```

Send the PDS dataset *HLQ.DOCS* as a.zip file attachment. We've specified the attachment name using the symbolic &JDATE (Julian date), so the resulting email would contain an attachment named –for example- *HLQ.DOCS.14090.ZIP.*The TRANSLATE operand is specified, so the PDS members will be readable on a PC.

```
//          EXEC PGM=OPSGS3B
//ATT1      DD DISP=SHR,DSN=HLQ.DOCS
//CONTROL  DD *
COMMAND    SEND
FROM       fred@acme.com
TO         betty@acme.com
Subject    Here is a sample zipped PDS.
ATTACH     DDN:ATT1 "HLQ.DOCS.&JDATE" TRANS ZIP
```

Using the TSO/ISPF dialogs

SMTP-Send provides an ISPF dialog to allow TSO/ISPF users to send SMTP mail. You can compose a message on the fly or you can send an existing data set as the message, and of course you can add attachments. Note that this facility does not allow TSO/ISPF users to receive e-mail. Online help is available to ISPF users by pressing Help (PF1) at any time.

SMTP-Send Main Menu

Normally you access the SMTP-Send Main Menu by issuing the command, '*TSO EX prefix.ISPCLIB(OPSMENU)*' or *OPSMENU* from the ISPF command prompt. Your installation may also make it available as an option on another panel. Check with your system administrator, to find out which way to invoke the dialogs. The Main Menu panel will be presented, providing the 3 choices as shown below. The first time the dialog is invoked, the user will first see the 'Settings' panel, as the ISPF environment is being set up, See CHANGING SMTP-SEND DEFAULTS.

```
                        SMTP-Send for z/OS Main Menu
 Option ===>


   0   Settings              Change your SMTP-Send for z/OS settings

   1   Compose a Message     Compose a new message and send it

   2   Send a File           Send a file




   User ...: wilycoyote@acme.com
   Userid .: ACME1WC
```

Composing a message

To compose a message on the fly, select option 1 from the Main Menu. Option 2 allows you to send an existing dataset as the message text. Both options also allow you to add other datasets as attachments. If you have specified files to attach, this is noted on the last line, as in this example. As shown below, Option 1 presents an addressing panel where you can provide the envelope information for your message. Option 2 (not shown) presents a similar addressing panel which allows you to specify the dataset name to be used as your message text.

```
      Attachments
 --------------------------------------------------------------------------
                        SMTP-Send for z/OS
 Command ===>
 Enter data and press ENTER to confirm values or END to exit
 Subject ... Overnight Report
                                                                   CC BCC
 Send to ... rr@acme.com
         ... bwayne@acme.com                                        X  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _
         ...                                                        _  _

 Reply-To(optional)..
 Files have been attached
```

Specify your recipients in the Send to field. You can specify up to 10 recipients, one per line. To make a user a cc: or bcc: recipient, type a non-blank character in the corresponding field on the right side.

OP/SS To specify SNADS addresses using the SMTP-Send ISPF panels, specify the SNADS values in parenthesis, i.e. (IBMUSER) or (IBMUSER.DSVHOST).

Adding Attachments

From either option 1 or 2, you can add attachments to your email. You can select the "Attachments" action bar choice, or type "Attach" into the command line. This will show the following panel, allowing you to add datasets to your attachment list. This example shows 3 attachments with various attach options. You add the first attachment by *Modify*'ing the first entry, which is initially blank. Then you can use either *Insert* or *Repeat* to create additional attachment entries on this list.

```
                  SMTP-Send for z/OS Attachment List              Row 1 of 3
 Command ===> _____    Scroll ===> CSR

  Choose attachments using the following line commands
   M Modify, I Insert,  D Delete,  R Repeat
 Press END to proceed.

 Sel   Data set Name

 _    'ACME1WC.DATA.VB(ASATXT)'
      / Translate    / ASA/MCC Control      RTF     ZIP

 _    /u/wiley/list
      / Translate      ASA/MCC Control      RTF    / ZIP

 _    'ACME1RR.DECOYS.HTM'
      / Translate      ASA/MCC Control    / RTF    / ZIP

 ***************************** Bottom of data ****************************
```

When you modify or insert an entry, you are presented with the 'Attach a Dataset' popup dialog, which allows you to type the dataset name, and specify some attachment options.
You can provide an optional newname for the attachment, but one would be automatically generated based upon the dataset name.
The other options correspond to options of the ATTACH control statement, and are described in, CREATE ATTACHMENT STATEMENTS. You can also hit PF1 for a popup help description of the field your cursor is on.
Dependent options cannot be selected unless the parent option has been selected. For example, if the Translate option is not selected, most of the others will be 'disabled', since they only apply when Translate is specified.
The existence of the dataset is verified before the entry can be saved to your attachment list.

```
+------------------------------------------------------------------------+
|                        Attach a Dataset                                |
| Command ===> _____                          |
|                                                                        |
| Press ENTER to update selections, END to save, CANCEL to quit.         |
|                                                                        |
| Dataset to Attach:   /u/wiley/list                                     |
|                      (HFS files must begin with '/')                   |
|                                                                        |
| Newname for Attachment (name seen by recipient):                       |
|                                                                        |
|    _____                                |
|                                                                        |
|    Attachment options. Enter '/' to select option.                     |
|                                                                        |
|    / Translate      _ Carriage Control (ASA or MCC)      _ Zip         |
|    _ Strip blanks      RTF: _      Pitch       P-Portrait/L-Landscape   |
|    / Convert LineFeed to CRLF                                          |
|                                                                        |
|                                                                        |
+------------------------------------------------------------------------+
```

When you have finished adding your attachment(s), and PF3/END out of the Attachment List panel, you will be returned back to the Addressing panel. If you had selected Option 1 in the Main Menu, then once you're done with the Addressing panel, you PF3/END to get to an ISPF Edit panel, where you can compose your message text, as shown below. This is standard ISPF Edit, with Addressing and instructions displayed in white, for your information (this information is not sent).

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 ------------------------------------------------------------------------
EDIT       ACME1WC.SMTPSEND.LISTDSN                           Compose Msg
Command ===>                                              Scroll ===> PAGE
****** **************************** Top of Data ****************************
======        To: rr@acme.com
======        Cc: bwayne@acme.com
====== Reply-To:
======  Subject: Overnight Report
====== ------------------------------------------------------------------
====== Datasets are attached
====== Enter SEND to send, ADDRESS to revise envelope or END to send later
====== Issue OPSHELP command for usage information
000001 Type your message text here...
''''''
''''''
''''''
''''''
''''''
''''''
''''''
```

When you have completed composing the message, enter the SEND command on the command line to send it. To save the message to send later, press PF3/End. If you want to quit without saving or sending the message, press Cancel.

Changing SMTP-Send defaults

The SMTP-Send Settings panel is normally selected from the SMTP-Send main menu. It may also be displayed the first time a user runs the dialog. This panel is prefilled with values specified by the SMTP-Send installer. To accept the values as they appear shown press Enter. You can change any of the highlighted values. The data set names used at your installation are shown for informational purposes. The CONTROL and EDITFILE datasets are allocated the first time a user invokes the dialogs, based on the naming template used by the installer. These are used to form the users email message. The CONFIGURATION and LOADLIB datasets are defined by the installer.

By default, your e-mail ID is created from your TSO userid and your installation's domain name. You can change this value as required. You can specify that any replies to your messages be sent to a different e-mail ID by specifying a value in the Reply-To field. You can override this selection at send time.

```
                Change your SMTP-Send for z/OS Settings
Command ===>
Press ENTER to save these values. Press END to quit
Sender's userid ....: ACME1WC
        e-mail ID .. wileycoyote@acme.com

Reply-To(optional)...

Document type ....... TEXT

Lines per page .. 66
VERSION:  60

SMTP-Send for z/OS Datasets
    CONTROL ........ 'ACME1WC.SMTPSEND.CONTROL'
    EDITFILE ....... 'ACME1WC.SMTPSEND.LISTDSN'
    CONFIGURATION .. 'SMTPSEND.R60.CONFIG'
    LOAD .......... 'SMTPSEND.R60.LOADLIB'


+-------------------------------------------------------------------------+
| Welcome to SMTP-Send for z/OS                                           |
| Since this is your first time using SMTP-Send for z/OS, we need to create |
| some datasets for you.                                                  |
| Press ENTER to proceed or END to quit                                   |
+-------------------------------------------------------------------------+
```

The message in red appears only on the users' first use of these dialogs. After pressing ENTER, the dialog will allocate the users CONTROL & EDITFILE datasets, and establish an ISPF profile for this application.

Using SMTP-Send under CICS

SMTP-Send includes a CICS interface that allows you to send e-mail from CICS applications. Your application program writes its e-mail request to a CICS Temporary Storage queue (TSQ) and then, via a CICS LINK, calls the SMTP-Send interface program, OPSGS3, passing a commarea.

The control statements are supplied in the same format as used in the SMTP-Send batch interface.

There are two formats of commarea that can be passed to OPSGS3. Both forms provide for the return of a character return code field to the caller and allow for the passing of control statements and data to OPSGS3 via TSQs. The WS-MERGDATA-TSQ field names a TSQ that contains the data provided in batch via the MERGDATA DDname used during HTML substitution.

The short commarea format requires the control statements and data to be combined into a single TSQ. The commarea is 20 characters in length, ending after WS-MTSQ-NAME. The WS-MTSQ-NAME TSQ id names the TSQ that contains both the control statements and data. This is analogous to providing all of the input in the Batch MESSAGE DD. Each record that is written to the MTSQ contains either a control statement or a block of the e-mail text. The control statements must appear first in the TSQ, terminated by a record with high-values (x'FF') in the first byte, and then followed by the e-mail text.

The long commarea format requires that you place a character '1' in the WS-VER field to indicate the long format. You can supply the control statements and data in a single TSQ like the short commarea format by placing the TSQ id in the WS-MTSQ-NAME field or separate the control statements and data into two TSQs. To supply the control statements in one TSQ and the data in another, write the control statements to the TSQ ID named in WS-CTSQ-NAME. The TSQ named in WS-MTSQ-NAME will only contain the data. No high-values record is required in this case.

The long format also allows you to supply From, To, Cc, Bcc, ReplyTo and Subject information directly via pointers. Each pointer points to a section of storage that contains the value to be used. The value must be terminated with a x'00' or low-value character. If any pointers are used to supply values, then the WS-CTSQ-NAME TSQ is not read and all required data must be supplied via pointers.

The names of the TSQ that you specify in the commarea must be unique to the transaction. The TSQs should be deleted after linking to OPSGS3 since OPSGS3 does not delete them. A common convention for creating a unique TSQ name is to use the CICS transaction name (EIBTRNID) followed by the CICS task number (EIBTASKN).

The commareas that can be passed to the OPSGS3 program are shown below:

```
    WORKING-STORAGE SECTION


       01  WL-COMMAREA.
          05 WL-RETURN-CODE       PIC X(4).
          05 WL-MERGDATA-TSQ      PIC X(8).
          05 WL-MTSQ-NAME         PIC X(8).


    WORKING-STORAGE SECTION


       02  WL-LONG-COMMAREA.
          05 WL-RETURN-CODE       PIC X(4).
          05 WL-MERGDATA-TSQ      PIC X(8).
          05 WL-MTSQ-NAME         PIC X(8).
          05 WL-VER               PIC X VALUE '1'.
    * REASON CONTAINS ONE OF THE FOLLOWING VALUES WHEN
    * WL-RETURN_CODE IS '0012'
          05 WL-REASON            PIC X(4).
             88 TO-INVALID          VALUE '0004'.
             88 CC-INVALID          VALUE '0008'.
             88 BCC-INVALID         VALUE '0012'.
             88 FROM-INVALID        VALUE '0016'.
             88 REPLYTO-INVALID     VALUE '0020'.
             88 SUBJECT-INVALID     VALUE '0024'.
          05 WL-CTSQ-NAME         PIC X(8).
    * FOLLOWING FIELDS CONTAIN POINTERS TO DATA
```

```
* TERMINATED BY A LOW-VALUE CHARACTER.
      05 WL-FROM-PTR         USAGE POINTER.
* MULTIPLE VALUES MAY BE SUPPLIED FOR TO, CC, BCC
* BY SEPARATING THEM WITH ';' CHARACTERS
      05 WL-TO-PTR           USAGE POINTER.
      05 WL-CC-PTR           USAGE POINTER.
      05 WL-BCC-PTR          USAGE POINTER.
      05 WL-REPLYTO-PTR      USAGE POINTER.
      05 WS-SUBJECT-PTR      USAGE POINTER.


   PROCEDURE DIVISION.
      EXEC CICS LINK
              PROGRAM('OPSGS3')
              COMMAREA(WL-COMMAREA)
              LENGTH(20)
              END-EXEC.
```

A sample COBOL program OPSTEST is included in INSTLIB. The OPST transaction uses this sample program to test your SMTP-Send installation under CICS.

OP/SS   Instead of OPSGS3, use program PIPGS3 (OfficePath/SNADS) or APIGS3 (RAPID). Use either the PIPT (OfficePath/ SNADS) or APIT (RAPID) transaction to test OP/SS under CICS. These transactions can only create the short form of commarea.

Note: Make sure that the CICS TCP/IP socket interface is active. Failure to do so may result in AEY9 abends. See IBM book IP CICS Sockets Guide, (SC31-8518), for information on starting the CICS Socket interface. (If you are using TCP/IP for MVS 3.2, see CICS TCP/IP Socket Interface Guide and Reference, (SC31-7131).)

HTML Mail Merge

Overview

HTML mail merge is an optional feature of SMTP-Send that allows you to send standard forms, such as invoices and purchase orders, as rich-text e-mail messages. Each e-mail message is custom built by merging application data into an existing HTML "form" or template. This is analogous to printing application data onto a pre-printed form.

How HTML mail merge works

There are two input streams to HTML mail merge, (1) MESSAGE which is the HTML template, and (2) MERGDATA which contains the application data. HTML mail merge substitutes the value of each variable in MERGDATA into the corresponding MESSAGE variable. When all variables are substituted the result is automatically passed to SMTP-Send for distribution as e-mail.

SMTP-Send example

Here is a sample of an SMTP-Send jobstream that uses the HTML mail merge feature. The CONTROL input stream specifies mail distribution information.

```
//OPSTEST  JOB TBS00
//STEP1    EXEC PGM=OPSGS3B
//STEPLIB  DD DISP=SHR,DSN=OPS.LOADLIb
//SYSPRINT DD SYSOUT=*
//CONTROL  DD *
* invoke HTML mail merge feature
COMMAND    MERGE
FROM       SALES@.TBSSOFT.COM
TO         SUPPORT@TBSSOFT.COM
INTYPE     TEMPLATE
//OPSCONFG DD DISP=SHR,DSN=OPS.CONFIG
//MESSAGE  DD DISP=SHR,DSN=OPS.HTML.TEMPLATE
//MERGDATA DD *
?COMPANY=THE ACME CORPORATION
?ADDRESS=56991 S.E. EXPRESSWAY
?CITY=GOTHAM CITY
?STATE=NY
?ZIPCODE=12345
?TELNO=(555) 555-1212
?AGREEMENT=66118580-0
?OWNCITY=01750-00
?REV%=
?DATEIN=06/16/99
?TIMEIN=14:49
?VIN=0006610729
?DATEOUT=06/15/99
?TIMEOUT=23:59
```

MERGDATA input

The MERGDATA input data set consists of a set of variable length records. Under CICS, the MERGDATA input is contained in a CICS TS queue whose name is specified by the MERGDATA parameter in the input commarea.

MERGDATA input can be formatted in two ways, using either keyword syntax or XML syntax.
The MESSAGE file is scanned for all occurrences of each keyword, and the corresponding data value is substituted. Keywords are not case-sensitive. End-of-file indicates the end of the input variables.

Syntax	Description
Keyword	Each record contains a single keyword and its data value. The keyword name may begin anywhere on the record, followed by an equal sign, followed by the data value. The text string to be substituted begins following the equal sign and ends at the last non-blank character.
XML	XML general entities indicate variables and their reference. The variable data in the MESSAGE file is denoted using XML general entity references. The MERGDATA input defines the entities in use through the XML declaration.

XML entities and their references are case-sensitive. To specify special characters use their ASCII values. For instance, use the string © (decimal) or © (hexadecimal) for the © character. For an example, see Batch merge of MIME template using keywords.

XML syntax is defined by a number of Internet Engineering Task Force (IETF) request for comments (RFCs.) The root RFC can be found at http://www.w3.org/TR/REC-xml#sec-physical-struct.

If the MESSAGE input stream contains multiple documents, you can supply multiple MERGDATA input by separating each set with a line containing ten or more x'FF' characters.

For example, this data input from the MERGDATA file:

```
?Title=Mr.
?User=Wile E. Coyote
?Address=1 Roadrunner Way, Steaming Desert
?Company=Acme Anvils
?Season=
FFFFFFFFFFFFFFFFFFFF
?Title=Ms.
?User=Daisy Duck
?Address=22 Skidoo, Anaheim
?Company=Disney & Co.
?Season=fall
```

... when substituted into this template in the MESSAGE file:

```
Dear ?Title ?User,
?Company is pleased to send our ?season newsletter to your address of
?Address.
```

... results in this output:

```
Dear Mr. Wile E. Coyote,
Acme Anvils is pleased to send our newsletter to your address of
1 Roadrunner Way, Steaming Desert.

Dear Ms. Daisy Duck,
Disney & Co. is pleased to send our fall newsletter to your address of
22 Skidoo, Anaheim.
```

MERGDATA-related control statements

Because the MERGDATA file is used only in conjunction with the HTML mail merge feature, you only need this file when you use the `COMMAND MERGE` control statement. When you use the TEMPLATE parameter, a single message is formatted as described in the previous section.

When you use the BATCH parameter, HTML Mail Merge sends a series of messages, each with its own unique addressing and variable substitution. In this case the MERGDATA file contains addressing and subject control

statements for each message, along with variable values for each message. The address and subject control statements are separated from each other using BOUNDARY statements.

SMTP-Send recognizes control statements in the MERGDATA file as summarized in the following table. The syntax is the same as for <u>distribution control statements</u>.

TO [+] *user*[*@domainname*]

Purpose: Specifies the e-mail address of the primary recipient.

CC [+] user[@domainname]

Purpose: Specifies the e-mail address of the secondary or informational recipient.

BCC [+]user[@domainname]

Purpose: Specifies the e-mail address of an additional recipient. BCC recipients are not included in the recipient lists of copies of the message sent to the primary and secondary recipients.

SUBJECT [+] subject

Purpose: Specifies the subject of the message.

FROM user[@domainname]

Purpose: Specifies the e-mail address of the sender.

REPLY-TO user[@domainname]

Purpose: Specifies the e-mail address to whom any reply will be sent.

Global and local values

Addressing and subject values may come from either the CONTROL or MESSAGE files, or from MERGDATA, or from a combination of these sources. When a value comes from CONTROL or MESSAGE it is considered to be global. This means the value is used in every message. When a value comes from MERGDATA it is considered to be local. This means that it is used only in the message with which it is associated.

A MERGDATA control statement may optionally have a + between the command and the local value. When a local value is specified without the + it completely replaces any corresponding global values. When a local value has a + that value is added to any corresponding global values.

When multiple global values are specified for TO, CC or BCC, any local TO, CC, BCC values will either replace all of the corresponding global values, or add to them.

Examples of HTML mail merge control statements

This section shows examples of SMTP-Send control statements that are used to perform typical HTML mail merge functions.
1. Single merge of MIME template using keywords
The MIME template in the MESSAGE file is merged with global data in the CONTROL file and local data in the MERGDATA file.
Input
//CONTROL DD * COMMAND   MERGE
TO   bar@foo.com
SUBJECT   Single Merge from ?company
FROM   your@name.com
INTYPE   TEMPLATE KEYWORD

//MESSAGE DD * MIME Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0005_01C034F6.F659E6E0"

------=_NextPart_000_0005_01C034F6.F659E6E0
Content-Type: text/plain

From:
?company name
?company address

//MERGDATA DD *
?company name = Acme Trading Ltd.
?company address = 66 Roadrunner Way

Output
MIME Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0005_01C034F6.F659E6E0"

------=_NextPart_000_0005_01C034F6.F659E6E0
Content-Type: text/plain

From:
  Acme Trading Ltd.
  66 Roadrunner Way

2. Batch merge of MIME template using keywords
This example also shows (a) how global variables are overridden or supplemented by local variables and (b) data substitution.
Input
//CONTROL DD *
BOUNDARY boundary
--boundary
COMMAND   MERGE
* global TO and SUBJECT
TO   bar@foo.com
*substitute order number in SUBJECT
SUBJECT   Order number ?order
FROM   your@name.com
INTYPE   TEMPLATE KEYWORD
//MESSAGE DD * MIME Version: 1.0
Content-Type: multipart/alternative;
boundary="----=_NextPart_000_0005_01C034F6.F659E6E0"
------=_NextPart_000_0005_01C034F6.F659E6E0
Content-Type: text/plain

?company name
?company address

//MERGDATA DD *
--boundary
* local TO replaces global TO
TO   sales@foo.com
* local SUBJECT replaces global SUBJECT
SUBJECT Your order received
--boundary--
?company name = Acme Trading Ltd.
?company address = 66 Roadrunner Way
--boundary
* local TO adds a recipient
TO+   imports@foo.com
* local SUBJECT adds "Now shipped"
SUBJECT+ Now shipped
--boundary--
?company name = Baker Exports Inc.
?company address = 73 Elm Street
* add order number
?order = 1212

Output
...
To: sales@foo.com

Subject: Your order received
MIME Version: 1.0
... From:
  Acme Trading Ltd.
  66 Roadrunner Way
...
To: bar@foo.com,imports@foo.com
Subject: Your order 1212
      Now shipped
MIME Version: 1.0
... From:
  Baker Exports Inc.
  73 Elm Street
...
3. Single merge of MIME template using XMLMERGE
XML entity values are substituted for general entities `&company_name;` and `&company_address;`.
Input
//CONTROL DD *
COMMAND   MERGE
TO   bar@foo.com
SUBJECT   Single merge
FROM   your@name.com
INTYPE   TEMPLATE XML
OUTTYPE   XMLMERGE

//MESSAGE DD * MIME Version: 1.0
... From:
  &company_name;
  &company_address;

   ...
//MERGDATA DD *
<!ENTITY company_name ="Acme Trading Ltd.">
<!ENTITY company_address ="66 Roadrunner Way">

Output
...
Subject: Single merge
MIME Version: 1.0
... From:
  Acme Trading Ltd.
  66 Roadrunner Way
...

4. Batch merge of MIME template using XMLMERGE
The input MIME template is merged with data in the MERGDATA file. The ASCII string `©` is used to represent the copyright character ©.
Input
//CONTROL DD * BOUNDARY bounds
--bounds
* invoke mail merge feature
COMMAND   MERGE BATCH
CC   sales@foo.com
SUBJECT   No subject
FROM   your@name.com
INTYPE   TEMPLATE XML
OUTTYPE   XMLMERGE
//MESSAGE DD * MIME Version: 1.0
... From:
  &company_name;
  &company_address;
...
  © Copyright 2001 &company_name;
...

//MERGDATA DD *

```
--bounds
TO      bar@foo.com
SUBJECT  Re: your order &order_no;
--bounds--
<!ENTITY company_name ="Acme Trading Ltd.">
<!ENTITY company_address ="66 Roadrunner Way">
<!ENTITY order_no ="123-4567">
--bounds
TO      post@foo.com
SUBJECT  Re: your order &order_no;
--bounds--
<!ENTITY company_name ="Beta Trading Inc.">
<!ENTITY company_address ="99 Roadrunner Way">
<!ENTITY order_no ="123-7654">
```

Output
```
...
To: bar@foo.com
Cc: sales@foo.com
Subject: Re: your order 123-4567
MIME Version: 1.0
... From:
  Acme Trading Ltd.
  66 Roadrunner Way
...
  © Copyright 2001 Acme Trading Ltd.
...
To: post@foo.com
Cc: sales@foo.com
Subject: Re: your order 123-7654
MIME Version: 1.0
... From:
  Beta Exports Inc.
  99 Roadrunner Way
...
  © Copyright 2001 Beta Exports Inc.
...
```

Preparing to use HTML mail merge

Note: Before you start to use the HTML mail merge feature make sure that SMTP-Send is installed correctly and that you can send plain text e-mails using it.

Your form designers will use HTML to replicate your existing forms. The designer must insert a placeholder, using either keyword or XML entity syntax, wherever you want application program data to be inserted. These placeholders must match the variable names or XML entity definitions in the MERGDATA input. Your application programmers and HTML designers will have to work together to agree on the names of these placeholders.

Usually, the HTML created on the PC will contain images or references to other files that reside on the workstation creating the HTML. The data that is sent by SMTP-Send must contain these files for the email to be fully portable. In addition, not all e-mail clients are capable of displaying the HTML format. Users using older e-mail clients or limited-function clients like pagers and PDAs may be unable to display the rich HTML.

SMTP-Send can solve this problem by sending the e-mail in multiple formats, plain text and full HTML. Using a no-charge PC e- mail client and web browser, you can transform the HTML template into a standalone format that contains images, other referenced files and alternate display formats.

Template creation

In order to use the HTML Mail Merge feature, you will need to set up one or more HTML templates. Since HTML development tools are not generally available on the OS/390 (MVS) platform, you will create these templates using standard PC-based tools then transfer them to MVS for use by SMTP-Send.

The procedure consists of these steps:

Edit HTML source
Test variable substitution
Create "flat" text stream
Add plain text version (optional)
Transfer text stream to OS/390

1. Edit HTML source

Create the HTML source using a standard editing tool such as FrontPage. Define the placeholder variables where required. Your file should look something like this:

```
<img border="0" src="tbslogo.gif" width=50 height=50>
<table>
<tr valign=top>
<td bgcolor=#fed70c>
<font size=2 face="Times New Roman">Direct All Inquiries To:</font>
<td colspan=2 bgcolor=#fed70c>
<font size=2 face="Times New Roman">Rental Agreement No:</font>
<td><font size=3 face="Times New Roman">?AGREEMENT</font>
<tr valign=top>
<td><font size=3 face="Times New Roman">?COMPANY</font>
<td bgcolor=#fed70c><font size=2 face="Times New Roman">Owning City No.</font>
<td bgcolor=#fed70c><font size=2 face="Times New Roman">Rev %</font>
<td bgcolor=#fed70c><font size=2 face="Times New Roman">Date/Time IN</font>
<tr valign=top>
<td><font size=3 face="Times New Roman">?ADDRESS</font>
<td><font size=3 face="Times New Roman">?OWNCITY</font>
<td><font size=3 face="Times New Roman">?REV%</font>
<td><font size=3 face="Times New Roman">?DATEIN ?TIMEIN</font>
<tr valign=top>
<td><font size=3 face="Times New Roman">?CITY, ?STATE ?ZIPCODE</font>
<td colspan=2 bgcolor=#fed70c>
<font size=2 face="Times New Roman">Vehicle No.</font>
<td bgcolor=#fed70c><font size=2 face="Times New Roman">Date/Time OUT</font>
<tr valign=top>
<td><font size=3 face="Times New Roman">?TELNO</font>
<td colspan=2><font size=3 face="Times New Roman">?VIN</font>
<td><font size=3 face="Times New Roman">?DATEOUT ?TIMEOUT</font></table>
```

In this example the placeholders begin with a question mark, "?".

2. Test variable substitution

You can test HTML substitution on your PC using the program SRCHREP, which is included with SMTP-Send. Find SRCHREP. EXE in host file OPS.vers.WIN32.SRCHREP.EXE and transfer it to your PC in binary format.

In addition to the HTML source file that you created in step 1. you will also need to create a MERGDATA file that contains test data.

Run SRCHREP, using the following syntax, in an MS-DOS command prompt window:

```
SRCHREP message mergdata [KEYWORD|XML]
```

SRCHREP reads the HTML source file, message, merges data from mergdata, and creates a new PC file, mergdata.out.html. Make sure to specify whether you are using keyword or XML syntax. You can view this file using a web browser to ensure that all variables are substituted correctly.

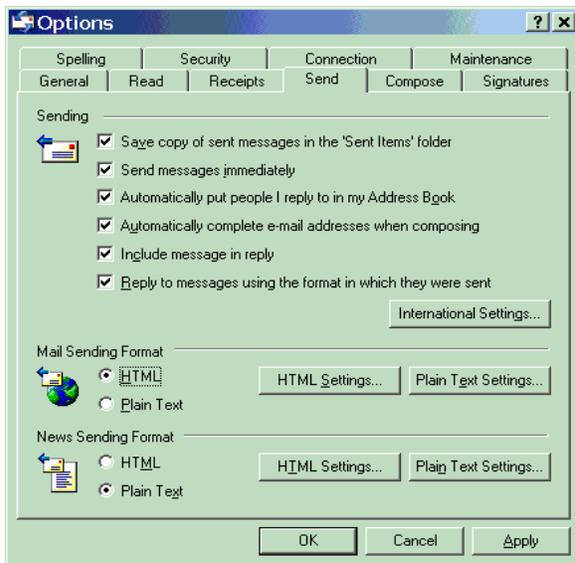### 3. Create "flat" text stream

The HTML source file that you created in step 1, will usually reference other files, for instance graphics in GIF or JPG format. In this step, called flattening, you will convert all of the constituent files of the web page into a single text stream. This stream contains all of the data that the recipient's e-mail client program will need to display the web page, including text, graphics and other referenced files.

To flatten HTML you need to use a product that will save the exact MIME datastream. You can use Microsoft's Outlook Express (OE) e-mail client and Internet Explorer 5.x+ web browser (IE). Both OE and IE programs are available at no cost from Microsoft's website. (You only have to install these programs on your PC. You don't need to make them your default e-mail client or web browser.)

Configure Microsoft Outlook Express

Option settings in Outlook Express determine the format of messsages. You must tell Outlook Express to send HTML. In addition, you can create a plain text version of the HTML to allow limited-function e-mail clients like pagers and cellphones to display the message.

1. Start Outlook Express.
2. On the Tools menu, click Options.
3. Click the Send tab.
4. Click HTML in Mail Sending Format.

To reduce the size of any images in your HTML.

1. Click the HTML Settings button.
2. Click Send pictures with messages.
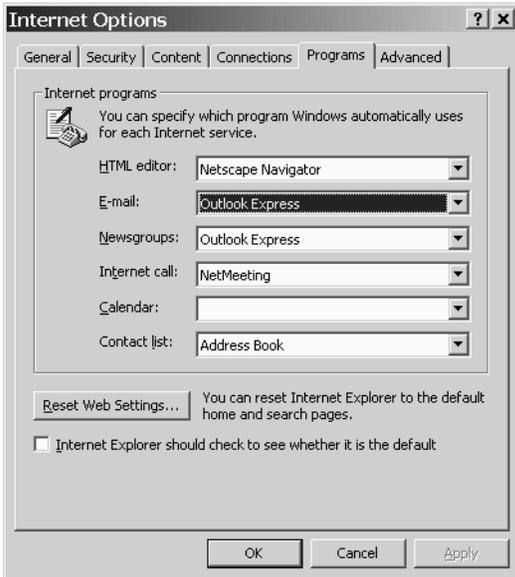3. In the Encode text using drop-down list select Base 64.

Click OK to save your settings, then click OK to save your options.
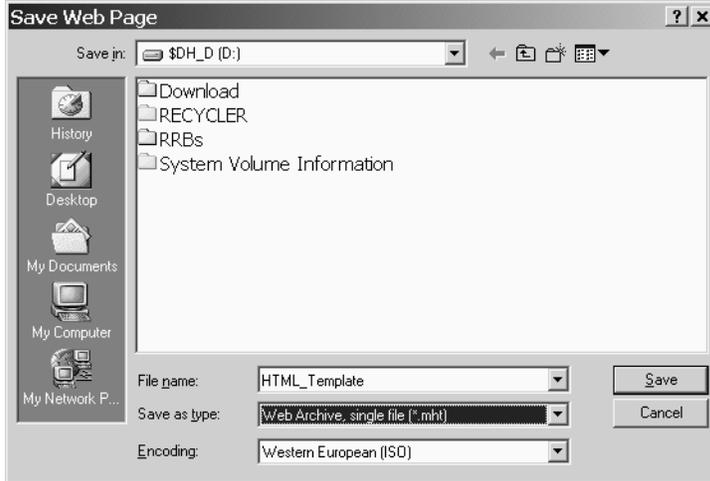Create alternate format with Microsoft Internet Explorer (IE)
Make sure that Outlook Express is the e-mail client that IE uses.
1. View the HTML template using IE.
2. Open the Tools menu, click Internet Options and click the Programs tab.
3. Make sure that the program specified in the E-mail drop-down list is Outlook Express.
4. Click OK.

Now save the HTML template in flattened format. While the HTML template is displayed in IE

1. Select the File menu and click Save As...
2. In the Save as type drop-down list select Web Archive, single file (*.mht).
3. Select the location and type the file name for the flattened file.
4. Click Save.

4. Add plain text version (optional)

While the HTML template is displayed in IE:

1. Select the File menu and click Save As...
2. In the Save as type drop-down list select Text file (*.txt).
3. Select the location and type the file name for the plain text file.
4. Click Save.
5. Open this text file with a text editor (e.g. Notepad) or word processor (e.g. Word). Reformat the text to make it readible. For example, if the text appears like this:

     Customer: ?NAME
     ?ADDRESS
     ?CITY, ?STATE ?ZIPCODE

you may want to reformat it like this:

     Customer: ?NAME
             ?ADDRESS
             ?CITY, ?STATE ?ZIPCODE

6. When you're done, select all of the text and copy it to the clipboard.
7. Using a text editor or word processor open the HTML template (file type mht) that you created in the previous step.
8. Locate the line `This is a multi-part message in MIME format..`
9. Select the entire line and replace it with the plain text from the clipboard.
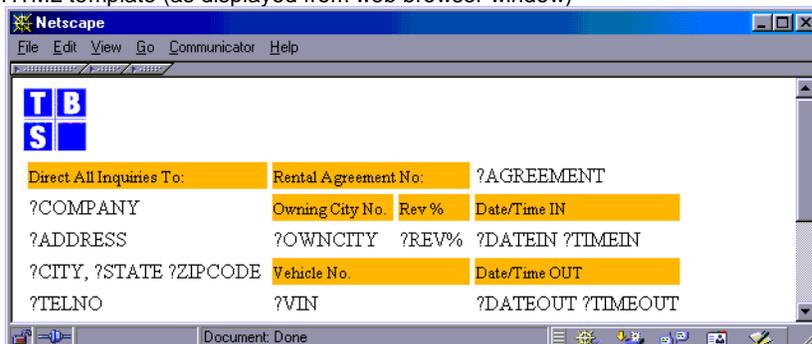10. Save the updated HTML template.

5. Transfer text stream to z/OS or OS/390

Using IND$file or FTP, transfer the text stream to the mainframe in text mode. SMTP-Send can now use this as the MESSAGE input file.
Example
Here is an example that shows how the HTML mail merge feature is used to create a Rental Agreement form.
HTML template (as displayed from web browser window)

Raw alternate format
```
From support@tbssoft.com Mon Jul 17 14:40:27 2000
Received: from [207.219.158.130] by hotmail.com (3.2) To: user@hotmail.com
Subject: template.html
Message-ID:
From: support@tbssoft.com
Date: Mon, 17 Jul 2000 17:37:31 -0400
MIME-Version: 1.0
Content-Type: multipart/related; boundary="=_related 0076AE7A8525691F_="

Direct All Inquiries To: ?COMPANY
                         ?ADDRESS
                         ?CITY, ?STATE ?ZIPCODE
                         ?TELNO
Rental Agreement No:     ?AGREEMENT
Owning City No.:         ?OWNCITY
Rev %:                   ?REV%
Date/Time IN:            ?DATEIN ?TIMEIN
Vehicle No.:             ?VIN
Date/Time OUT:           ?DATEOUT ?TIMEOUT
--=_related 0076AE7A8525691F_=
Content-Type: text/html; charset="us-ascii"

<br><font size=3 face="Times New Roman"><br><br></font>
<!--img src=cid:_1_1AA4000025CC0076AE788525691F-->
<img src="tbslogo.gif" height=50 width=50>
<table>
<tr valign=top>
<td bgcolor=#fed70c><font size=2 face="Times New Roman">Direct All Inquiries To:</font>
<td colspan=2 bgcolor=#fed70c>
<font size=2 face="Times New Roman">Rental Agreement No:</font>
<td><font size=3 face="Times New Roman"> ?AGREEMENT </font>
...
<tr valign=top>
<td><font size=3 face="Times New Roman">?TELNO </font>
<td colspan=2><font size=3 face="Times New Roman">?VIN</font>
<td><font size=3 face="Times New Roman">?DATEOUT ?TIMEOUT</font></table>
--=_related 0076AE7A8525691F_= Content-Type: image/gif
Content-ID: <_1_1AA4000025CC0076AE788525691F> Content-Transfer-Encoding: base64
R0lGODdhZABkAPcAAAAAAGgAAJkAAMAAAOEAAP8AAABoAGhoAJloAMBoAOFoAP9oAACZAGiZAJmZ
AMCZAOGZAP+ZAADAAGjAAJnAAMDAAOHAAP/AAADhAGjhAJnhAMDhAOHhAP/hAAD/AGj/AJn/AMD/
... THR8zTWQCHbLGz3PNwX4ss4jyilebGNDE/039R8lpfJpeLq+XFtK54zyhif+qv5mIkfbIE9PqD2C
BU7+6O3j/v09rNy0ZEyjN1bNVoAGaBABAQA7
--=_related 0076AE7A8525691F_=--
```
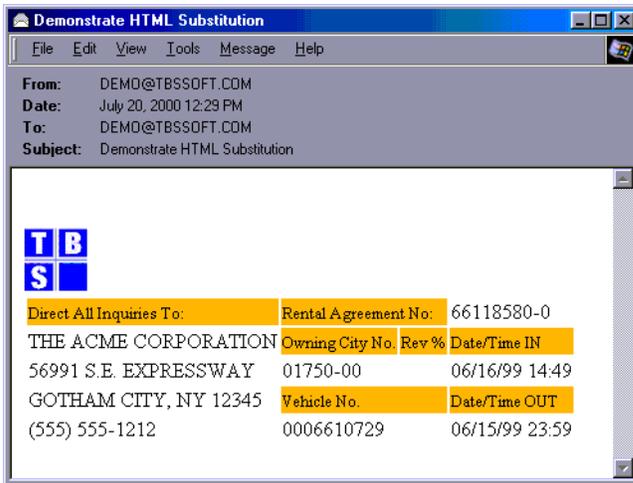
MERGDATA input
```
?COMPANY=The Acme  Corporation
?ADDRESS=56991 S.E. Expressway
?CITY=Gotham City
?STATE=NY
?ZIPCODE=12345
?TELNO=(555) 555-1212
?AGREEMENT=66118580-0
?OWNCITY=01750-00
?REV%=
?DATEIN=06/16/99
?TIMEIN=14:49
?VIN=0006610729
?DATEOUT=06/15/99
?TIMEOUT=23:59
```

Message e-mailed by SMTP-Send when viewed in Outlook Express

Message e-mailed by SMTP-Send as viewed in plain text

```
Direct All Inquiries To: The Acme Corporation
                         56991 S.E. Expressway
                         Gotham City, NY, 12345
                         (555) 555-1212
Rental Agreement No:     66118580-0
Owning City No.:         01750-00
Rev %:
Date/Time IN:            06/16/99 14:49
Date/Time OUT:           06/15/99 23:59
Vehicle No.:             0006610729
```

OfficePath CARDFILE control statements

OP/SS This table summarizes those OfficePath and RAPID control statements that are processed by OP/SS. Other OfficePath and RAPID control statements have no equivalent in SMTP and are ignored by OP/SS. For a complete description of OfficePath and RAPID control statements and their syntax, refer to the documentation for those products.

Control Statement	Notes
COMMAND SEND	NOTE, DOCUMENT and MESSAGE are acted upon. PROFILE and ACK are ignored.
DCA LEVEL value	Values 1403 (the default), 0002, 0011 and 0014 are recognized as Text, FFT-DCA, RFT-DCA and Binary, respectively. Sub-parameters ASA, CHARS, LINES and MARGIN are recognized.
MESSAGE text	
RECIPIENT userid [node]	Although OfficePath and RAPID support a maximum of 50 recipients, OP/SS is limited only by the amount of available storage.
REPLY Y|N	
TO USER userid [node]	1. If REPLY is Y, OP/SS expects a TO USER parm to follow. OP/SS uses the TO USER to generate an SMTP Reply-To. 2. Unlike OfficePath or RAPID, OP/SS does not return any confirmation of delivery (COD) or receipt acknowledgement (RA) status to the sender of a distribution.
SUBJECT text	
USER NAME userid [node]	

Messages and codes

Messages

Abnormal termination of SMTP-Send will be described with a non-zero return code and an accompanying message.

Authorization code is invalid. Reverting to demo mode
The product authorization code specified in the Configuration file is either invalid or missing. SMTP-Send processing degrades to demonstration mode.

Authorization expired date. *n* days left in grace period
The product authorization expired on the specified date. SMTP-Send will continue to run for *n* days.

`OP/SS` Check CARDFILE for unknown entry in SEND COMMAND
An invalid parameter was specified for the SEND COMMAND. Correct the CARDFILE data and rerun the job.

Commarea pointer is NULL
The CICS API program was started without a commarea. Correct the calling program and rerun.

Demonstration Mode. Not for productive use
This message informs you that your authorization to use SMTP-Send has expired.

`OP/SS` DGN.DEN not in CONFIG file
The SNADS distribution value DGN.DEN could not be mapped using values in the CONFIG file. Correct the CONFIG file and rerun the job.

DynAlloc error *msg*
msg will contain information returned by Dynamic Allocation. Correct the error described and rerun the job

DynAlloc keyword string too long
The keywords specified for the TESTOUTDSN data set are longer than 256 bytes in length. Correct the TESTOUTDSN statement and rerun the job.

Error accessing file CICS Journal 2, RC=43, Journal ID error
The program was unable to write to the CICS journal. More information is written to the MSGUSR output log.

Error accessing file LOGFILE
The program was unable to write to the LOGFILE file.

Error accessing file SMF dataset, RC=16, Not active
The program was unable to write the MVS SMF dataset.

Error writing data to file OUTPUT
The program was unable to write the specified data to the OUTPUT file. This message will be preceded by messages from the library I/O routine describing the problem in detail. Correct the error and rerun the job.

*ERROR * 'data' CONFIG data is invalid. It is ignored
An invalid CONFIG data statement was encountered. Correct the input and rerun the OPSCONFG utility.

EXEC CICS FAIL: FIBFN=*value*,EIBRESP=*value*,EIBRESP2=*value*
An unexpected CICS error was encountered. Contact Technical Support.

LRECL of *n* is too short for file OUTPUT
The logical record length for the preallocated file OUTPUT is too short to contain the data supplied. Correct the allocation and rerun the job.

Mandatory statement *value* is missing
The statement *value* must be included in the CONTROL data. Correct the job and rerun.

No attachments are possible in NOMIME state
You cannot add attachments when the OUTTYPE is NOMIME.

No records in file *DDNAME*

The named file is empty. Correct the JCL and rerun the job

OPSCONFG data for Key *key* is missing
A required control statement, key, was not supplied. Correct the input and rerun the OPSCONFG utility.

OPS126E FAILURE RELAYING TCPIP DATA
This text is written via WTO then SMTP-Send is unable to contact both the primary and secondary SMTP server to send a message via TCP/IP.
OP/SS The OPS prefix may be replaced with PIP when running OP/SMTP-Send with program PIPGS3B or the SoftSwitch Application Toolkit.

Product Authorization expired *date*. Reverting to demo mode
Your authorization to use SMTP-Send expired on the specified *date*. SMTP-Send processing degrades to demonstration mode.

Required DDname *ddname* missing
A required DDNAME is missing. Correct the JCL and rerun the job.

OP/SS Required module *module* missing
The OfficePath module PIPSIN or the RAPID module RAPIDIN is required by OP/SMTP-Send. Add the required module to the STEPLIB DD and rerun the job.

Statement *value* contains invalid data
An unrecognized control statement was encountered. Correct the job and rerun.

Statement *value* unsupported
An unrecognized control statement was encountered. Correct the job and rerun.

Statement sequence error *value*. First statement must be COMMAND
The control statement value was encountered before a COMMAND control statement.

Storage allocation failure in program
The program program was unable to allocate memory. Increase the region size and rerun the job.

This message sent by SMTP-Send in Demonstration Mode
This message text is added to each message sent by SMTP-Send. The demonstration authorization mode restricts SMTP-Send to sending only the first line of the message text.

TSQ ID *tsqname* not found
The CICS TS Queue name supplied in the CICS API commarea could not be found. Correct the calling program and rerun.

Unable to open file *DDname*
The address validation exit denied the use of name as an e-mail address.

Use of e-mail address *name* denied by exit
The address validation exit denied the use of name as an e-mail address.

Value of CHARS - MARGIN (*value*) greater than LRECL of *lrecl*
The logical record length of the OUTPUT file is less than the message text record length calculated from the number of characters per record minus the left margin. Correct the allocation and rerun the job.

Value returned by exit is longer than 128 characters
The address validation exit attempted to replace an e-mail address with a valid longer than 128 characters in length. Values returned by the exit are limited to 128 characters.

WARNING type value *'data'* already in file, Record ignored
A duplicate data statement was encountered. Correct the input and rerun the OPSCONFG utility.

Return codes from OPSGS3B (Batch)

Code	Meaning
0000	Processing completed normally.
0008	Internal Error - refer to additional messages issued
0016	Authorization code has expired
0020	Expected DDname missing
0032	LDAP lookup failed to connect to LDAP Server or Address failed validation
0100	Unable to access MESSAGE.
0101	No records in MESSAGE.
0103	Error reading MESSAGE.
0104	Error processing message text. Message text was not found when expected.
0105	Internal logic error. Contact Technical Support for assistance.
0106	Control statement sequence error. The COMMAND control statement must be the first control statement. Other control statements may appear in any order.
0107	Control statement value blank or invalid.
1046	OPSCONFG file cannot be found
02xx	Control statement data invalid. The control statement xx contains invalid data.
03xx	Control statement parameter invalid. The control statement parameter xx contains invalid data.
04xx	Control statement parameter value invalid. The control statement parameter xx contains an invalid or inappropriate value.
05xx	Mandatory control statement missing. Control statement xx is required but was not supplied.
07xx	LDAP error. A sender or recipient value is in error or is missing. No message was sent.
1030	Error reading CONTROL
1046	Could not OPEN the OPSCONFG file.

Return codes from OPSGS3 (CICS)

Code	Meaning
0000	Processing completed normally.
0008	Internal Error - refer to additional messages issued
0016	Authorization code has expired
0032	LDAP lookup failed to connect to LDAP Server or Address failed validation
0104	Error processing message text. Message text was not found when expected. Check the format of the input TSQ.
0105	Internal logic error. Contact Technical Support for assistance.
0106	Control statement sequence error. The COMMAND control statement must be the first control statement. Other control statements may appear in any order.
0107	Control statement value blank or invalid.
02xx	Control statement data invalid. The control statement xx contains invalid data.
03xx	Control statement parameter invalid. The control statement parameter xx contains invalid data.
04xx	Control statement parameter value invalid. The control statement parameter xx contains an invalid or inappropriate value.
05xx	Mandatory control statement missing. Control statement xx is required but was not supplied.
1044	Error reading input TSQ. A message providing the CICS command EIBRCODE and EIBRESP values is also written to the CICS message log.
1046	Could not OPEN the OPSCONFG file.

The xx in the above codes identifies the control statement keyword according to this list:

Code	Statement
00	COMMAND
02	SUBJECT
03	INTYPE, OUTTYPE
04	TO, CC, BCC
05	FROM

OP/SS OP/SMTP-Send return codes are the same as those that would be returned by OfficePath or RAPID under similar circumstances. Refer to the Problem Determination Guide for the appropriate product for additional information.