



Installing and Using SMTP-Send™ OfficePath/SMTP™ Release 5.2

This edition applies to [SMTP-Send for OS/390 and z/OS](#) Release 5.2 and to [OfficePath/SMTP-Send for OS/390 and z/OS](#) Release 5.2. It also applies to subsequent releases until otherwise indicated in new editions. ([Change history](#))

Using SMTP-Send

[Introduction](#)

[Sending messages](#)

[Using the ISPF dialogs](#)

[Using SMTP-Send under CICS](#)

[HTML mail merge](#)

Installing SMTP-Send

[Installing SMTP-Send](#)

[Configuring SMTP-Send](#)

[Customizing user exits](#)

[Changing message text](#)

[Customizing ISPF panel support](#)

[Using TBS SMTP Spool Server](#)

Appendices

[Messages and codes](#)

[Using LDAP](#)

[Log record format](#)

[Dynamic allocation keywords](#)

[OfficePath CARDFILE control statements](#)

TBS Software Inc.

8300 Woodbine Ave, Suite 301

Markham, Ontario, Canada L3R 9Y7

Tel: +1 905 940 9373 • Fax: +1 905 940 9376

<http://www.tbsoft.com/smtpsend52/smtpsend52.html>

Copyright © 1999-2005 TBS Software Inc. All Rights Reserved.

The software described in this publication is furnished under license and may be used or copied only in accordance with the terms of the license agreement. 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.

Table of Contents

Introduction	1	
Overview		1
Highlights		1
Benefits		1
Other books you may need		1
OP/SS Overview of OfficePath/SMTP		2
Additional Highlights		2
Additional Benefits		2
Other books you may need		2
Soft*Switch Application Toolkit compatibility		2
Sending messages	3	
SMTP-Send input		3
Simple SMTP-Send jobstream		3
Distribution control		3
Message text		4
Control Statements		4
Control statement syntax		4
Distribution Request Type Statements		4
Content Description Statements		5
Create Attachment Statements		7
Batch/TSO:		7
CICS:		7
Formatting Control Statements		8
Cascading Style Sheet support		9
PDF Creation control statements		9
Symbolic Substitution		10
Control statement examples		11
1. Simple text message with no attachments		11
2. Message with two attachments		11
3. Multiple sends in one batch job		11
4. Multiple sends from CICS		11
CICS TSQ input.....		11
5. Setting your own start and end boundaries		12
6. Using your own boundary		12
Calling SMTP-Send directly via LINK or ATTACH		12
Overriding the system configuration file (OPSCONFG)		13
Notification of job status (OPSNOTFY)		13
Logging distributions		14
Logging distributions directly from SMTP-Send		14
Sending messages with OfficePath/SMTP-Send		14
Sample OfficePath/SMTP-Send jobstream		14
OfficePath/SMTP-Send Extensions		15
Native SMTP addressing.....		15
IBMMail addressing compatability.....		15
Soft*Switch Application Toolkit Feature.....		16
Using the TSO/ISPF dialogs	17	
Composing a message		17
SMTP-Send Main Menu		17
Changing SMTP-Send defaults		18
Using SMTP-Send under CICS	19	
HTML Mail Merge	21	
Overview		21
How HTML mail merge works		21
SMTP-Send example		21
MERGDATA input		21
MERGDATA-related control statements		22
Global and local values		22
Examples of HTML mail merge control statements		23
1. Single merge of MIME template using keywords		23
2. Batch merge of MIME template using keywords		23
3. Single merge of MIME template using XMLMERGE		24
4. Batch merge of MIME template using XMLMERGE		24
Preparing to use HTML mail merge		25
1. Edit HTML source.....		25
2. Test variable substitution.....		26
3. Create "flat" text stream.....		26
4. Add plain text version (optional)		28
5. Transfer text stream to z/OS or OS/390.....		28
Example		28
HTML template (as displayed from web browser window).....		28
Installing SMTP-Send	30	
Requirements		30

Distribution package contents	30
Installation data sets	30
Authorization codes	30
Installing SMTP-Send	30
Before you start the installation	30
To install SMTP-Send	31
CICS installation	31
APF authorization	31
Logging distributions	32
Logging from batch to SMF	32
Logging from batch to a sequential file	32
Logging from CICS	32
Soft*Switch Application Toolkit Feature	32
If you have never used OfficePath previously	32
Testing the Application Toolkit Feature	32
Installing SMTP-Send in production without JCL changes	33
Configuring SMTP-Send	34
Creating the OPSCONFG file	34
Syntax conventions	34
Creating the OPSCONFG VSAM data set	34
CONFIG control statements	34
Specifying the default domain name	34
Setting the authorization code	35
Specifying the SMTP Server timezone offset	35
Specifying observance of Daylight Savings Time	35
Specifying Daylight Savings Time rules	35
Changing the SMTP Server information	35
Controlling the SMTP transmission	37
Highlighting Notices in Microsoft Outlook	37
Redirecting output	37
Handling addressing errors	38
Enabling user exits	39
Controlling the look of the message	40
Controlling conversion to Adobe PDF format	42
Debugging SMTP-Send operation	43
Adding a standard Notice to each message	43
Distribution list control statements	44
Logging distributions	44
LOGGING	44
LOG_JNUM	45
SMF_TYPE	45
NODES and USERS address mapping control statements	45
SSWDA OPSCONFG control statements	46
SSWMSGEncode	46
SSW_Header_Compatibility	46
SSWDA Program customization	46
Customizing user exits	47
Performing your own distribution list processing	47
Parameter List	47
Start of DLDATA Extension	47
OP/SS Performing your own address mapping	51
Changing SMTP-Send messages	54
Translating messages into a different language	54
How SMTP-Send messages are structured	54
The message definition macro #TBSSTRD	54
Coding message text	54
Quotation marks	54
Insertion points	54
Long messages	55
Special characters in languages other than English	55
Customizing ISPF panel support	56
Installing the SMTP-Send dialogs using standard ISPF libraries	56
Customizing the SMTP-Send dialogs	56
Using the TBS SMTP Spool Server	57
Overview	57
Configuration Parameters	57
Using the SMTP Spool Server	57
ATSIGN	57
BATCH_AUTHCODE	57
DEBUG	58
DEFAULT_DOMAIN	58
ERROR_MSG_FROM_NAME	58
ERROR_MSG_TO_NAME	58
FAIL_ON_ERROR	58

FIXSPOOLLEADINGDOTS.....	58
LOGGING.....	58
RELAY_RETRY_COUNT.....	58
RELAY_RETRY_DELAY.....	58
RELAY_LOG_DETAILS.....	58
RELAY_SMF_TYPE.....	58
REMOVETRAILINGBLANKS.....	59
SMTP_RELAY_ADDRESS.....	59
SMTP_RELAY_ADDRESS_ALTERNATE.....	59
SMTP_SERVER_CLASS.....	59
SMTP_SERVER_NAME.....	59
SMTP_SERVER_NODE.....	59
USECUSTOMCODEPAGE.....	59
WTO_onTCPFailure.....	59
Installation Considerations.....	59
Logging.....	59
Running under MVS.....	60
EXEC runtime parameter.....	60
Console Operator Commands.....	60
Optional JCL statements.....	60
Messages and codes.....	60
Return Codes.....	60
User Abends.....	61
Messages and codes.....	62
Messages.....	62
Return codes from OPSGS3B (Batch).....	63
Return codes from OPSGS3 (CICS).....	64
Using LDAP.....	65
Overview.....	65
Expansion of distribution lists.....	65
E-mail address lookup.....	65
Requirements and limitations.....	65
LDAP client API.....	65
MVS Batch/TSO.....	65
CICS.....	65
Testing your LDAP environment.....	65
Activating LDAP lookups.....	66
CICS.....	67
Additional LDAP Configuration Statements.....	67
LDAP_OpenRetry {3 count}.....	67
LDAP_OpenDelay {5 seconds}.....	67
LDAP_Timeout {60 seconds}.....	67
LDAP_Password value.....	67
Calling LDAP from your lookup, validation or data exit.....	67
Calling the LDAP lookup program.....	67
Passing Data to LDAP Lookup Program.....	68
Searching using LDAP Lookups.....	69
Migrating from a release prior to Release 5.2.0.....	70
Migrating custom LDAP exits.....	70
Customers calling SMTP-Send's LDAP exits from their own lookup, data or address validation exit.....	70
Supplementary LDAP reading.....	70
Log record format.....	71
Overview.....	71
Offset/length values.....	71
Fixed length fields.....	71
Variable length fields.....	71
Annotated record user data format.....	72
Dynamic allocation keywords.....	73
OfficePath CARDFILE control statements.....	74
Change history.....	75
R5.2.....	75
Enhancements.....	75
Fixes.....	75
R5.1.....	75
Enhancements.....	75
Improvements.....	75
Fixes.....	75
R5.0.....	76
Enhancements.....	76
Miscellaneous Documentation Updates.....	77
Fixes.....	77
R4.21.....	78



Introduction

Overview

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

Title

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. 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 Icom Solutions' (now Keane UK) 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 Icom Solutions' 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

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 may be [logged](#) for message tracking.

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.

You can also use SMTP-Send to automatically send a message about the completion status of a job using the [OPSNOTFY](#) program described below.

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 with OfficePath/SMTP-Send](#) for additional information.

SMTP-Send input

SMTP-Send uses control statements to indicate 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 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.ver.s.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.ver.s.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 contents is in the MESSAGE dataset. For example, you could modify your overnight report jobs to create their output in a temporary dataset which could then be used as the MESSAGE dataset. Global settings that apply to all requests are specified in [OPSCONFG](#), the SMTP-Send configuration dataset. You can use your own configuration dataset by following the instructions in the [Overriding the system configuration file](#) section.

Distribution control

Control statements are supplied to SMTP-Send in one of two ways:

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 OS/390 Batch and TSO. If you want to explicitly indicate the use of card image, add PARM='C[ONTROL]' to the EXEC statement in your SMTP-Send JCL stream.

With the text The control statements are stored along with the message text in DDname MESSAGE. To specify this mode, add PARM='M[ESSAGE]' 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. ...

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 create the output text so that pagination is preserved 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.

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. The operands must begin in column 11. Each operand, if there is more than one, is separated by one or more blanks.

A control statement that begins with an asterisk (*) is a comment and is not processed. Likewise, blank control statements are ignored.

All commands must be in upper-case. Operands are case-insensitive. Line numbers are ignored if the first non-comment control statement contains line numbers.

This table summarizes distribution control statements that SMTP-Send recognizes. See also [HTML Mail Merge control statements](#) if you use this optional feature.

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 whom any error messages regarding this email 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.

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 characters 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

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

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 file name is derived from the MVS DSNName 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 as is. 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.
SYMBOLS	Symbolic variables prefixed by & are recognized and replaced in the message text, subject and filenames. Refer to Symbolic Substitution for more information on symbols.
NOSYMBOLS	Symbolic variables prefixed by & are not recognized or replaced
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, "<", "&" and ">" respectively. 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>"
EMPTY_ATT	For INTYPE BINARY, PDF, RTF or OUTTYPE ATTACHMENT messages only, the message is sent as an attachment even when it contains no data. Use this parameter to override the OPSCONFG <code>SendEmptyAttachments No</code> control statement. NOEMPTY_ATT For INTYPE BINARY, PDF, RTF or OUTTYPE ATTACHMENT messages only, the message is not sent as an attachment when it contains no data. Use this parameter to override the OPSCONFG <code>SendEmptyAttachments Yes</code> control statement.

Create Attachment Statements

Batch/TSO:

```
ATTACH { DDN:ddname | DSN:dsname | HFS:pathname } [newname]
      [TRanslate [ASA|NOASA] [RTF[(pointsize[-(L|P)]) ] | NORTF]
      [RTB|LTB] [LF2CrLf | NOLF] [SEND_EMPTY | NOSEND_EMPTY] ]
      [ZIP]
      [PDF[(Blas() PAper() Width( ) Height() Pltch() Orient( ) )]]
      [PDF[(TM() BM() LM( ) RM( ) )]]
```

CICS:

```
ATTACH { TSQ:tsqname | TDQ:tdqname } [newname] [TRANSLATE [ASA|NOASA]
      [RTF[(pointsize[-(L|P)]) ] | NORTF] [RTB | LTB] [LF2CrLf | NOLF]
      [SEND_EMPTY | NOSEND_EMPTY] ]
      [PDF[(Blas() PAper() Width( ) Height() Pltch() Orient( ) )]]
      [PDF[(TM() BM() LM( ) RM( ) )]]
```

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.
 TDQ The CICS TDQ name of the file to be attached.

newname The name of the attached file as it appears in the message.
 Notes:

1. If newname could be confused with one of the other ATTACH keywords enclose the value in double quotes.
2. If newname is not specified, then the MVS data set, HFS file, CICS TSQ or CICS TDQ name is used.
3. The case of the newname value is preserved.
4. 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.

TRANSLATE The input file is translated from the EBCDIC code page to ASCII before being sent.
 ASA - any printer 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. ASA is not supported for ZIPped attachments.
 NOASA - ignores the printer control specification in the input file's DCB.

RTF The message 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 default pitch and page orientation specifying subparameters to RTF. Pointsizes indicates the size of size of the font in points, **L** indicates Landscape orientation and **P**, Portrait orientation. If not specified, the default pitch is obtained from the PITCH control statement or configuration statement. e.g `RTF(10)` indicates 10 point (Landscape is defaulted). `RTF(12-P)` indicates 12 point Portrait.

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	Trailing blanks are removed from translated input lines. RTB is an abbreviation for <code>RemoveTrailingBlanks</code> . LTB is an abbreviation for <code>LeaveTrailingBlanks</code> which instructs SMTP-Send to not remove trailing blanks. These keywords are used to override the <code>OPSCONFG RemoveTrailingBlanks</code> and <code>CONTROL RTB & LTB</code> settings.
LF2crlf	LineFeed characters encountered in HFS text files are replaced with a CRLF sequence. <code>NOLF2crlf</code> prevents this substitution. These keywords are used to override the <code>OPSCONFG ReplaceLFwCRLF</code> and <code>LF2CRLF CONTROL</code> settings. Non-HFS files are not affected by this keyword.
ZIP	Indicates that the attached file is to be ZIP compressed. A basic ZIP compression feature that is compatible with <code>PKZIP/PKUNZIP</code> is included in the SMTP-Send software.
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.
SEND_EMPTY	For <code>INTYPE BINARY</code> , <code>PDF</code> , <code>RTF</code> or <code>OUTTYPE ATTACHMENT</code> messages only, the attached file is sent as an attachment even when it contains no data. Use this parameter to override the <code>OPSCONFG SendEmptyAttachments No</code> control statement. <code>NOSEND_EMPTY</code> For <code>INTYPE BINARY</code> , <code>PDF</code> , <code>RTF</code> or <code>OUTTYPE ATTACHMENT</code> messages only, the attached file is not sent as an attachment when it contains no data. Use this parameter to override the <code>OPSCONFG SendEmptyAttachments Yes</code> control statement.

Formatting Control Statements

IMPORTANCE {L|O|w | N|O|r|m|a|l | H|i|g|h}

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.

`L|O|w` signifies the message is of low importance
`N|O|r|m|a|l` signifies the message is of normal importance
`H|i|g|h` signifies the message is of high importance

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.

Example: `LTB`

PITCH {10 |value}

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

ORIENT {10 |value}

Purpose: Specifies the orientation used for RTF attachments when no value is specified in the RTF subparameter.

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 `OPSCONFG RemoveTrailingBlanks` setting to force the removal of trailing blanks from text input

files.

Example: RTB

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[White]},{foreground_color[Black]}

Purpose: Specifies or overrides background and foreground colors when Cascading Style Sheets are used to format HTML text. Refer to [Configuring](#) for the color options.

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 `` 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. 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. Text is rendered in 12pt Courier, with page length and width calculated from the lines per page and characters per line. Top, bottom, left and right margins are all set to zero.

Print causes SMTP-Send to use the additional PDF defaults described below or their overridden values.

PDFBOTM {inches | millimetersmm}

Purpose: Specifies bottom margin

Default: PDFBOTM 0

PDFHEIGHT {inches | millimetersmm}

Purpose: Specifies paper height

Default: PDFHEIGHT 11

PDFLM {inches | millimetersmm}

Purpose: Specifies left margin

Default: PDFLM 0

PDFORIENT {Portrait | Landscape}

Purpose: Defines orientation of the page when printed.

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

Purpose: Specifies common page sizes. Height and width may be overridden by PDFWIDTH and PDFHEIGHT statements.

PDFPITCH {12| pointsize}

Purpose: Specifies size of the text font in points.

PDFRM {inches | millimetersmm}

Purpose: Specifies right margin
Default: PDFRM 0

PDFTOPM {0 |inches | millimetersmm}

Purpose: — Specifies right margin
Default: PDFTOPM 0

PDFWIDTH {inches | millimetersmm}

Purpose: Specifies paper width
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 PDFWIDTH 216mm

Examples:

```
PDFBIAS P
PDFPITCH 9
PDFORIENT L
PDFWIDTH 8.5
PDFHEIGHT 11
PDFTOPM 0.25
PDFBOTM 0.25
PDFLM 0.25
PDFRM 6.35mm
```

Specifies the standard page layout for a laser printer. with letter paper stock (8.5 in x 11 in) , 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 12
PDFORIENT L
PDFPAPER A4
```

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
PDFWIDTH 210mm
PDFHEIGHT 297mm
```

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

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 a offset (positive or negative) to be used in calculated the displayed date from the current date.

&JOB	Jobname (1-8 characters)
&JOBID	JobID (1-6 characters) i.e J12345 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 (yyymmdd 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 mmddy 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


```
...
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
```

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
Page 12

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.

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 provide an override file, when testing a new e-mail enabled application.

For information on the contents of the OPSCONFG file, and control statements syntax, see [Configuring](#).

Notification of job status (OPSNOTFY)

The OPSNOTFY program makes it easy to notify a user by e-mail when their OS/390 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 customise 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. Default 'S' from CUSTSNDT

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

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 JESMSGLOG 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.

Logging distributions

SMTP-Send logs the sending of e-mails using an external dataset or the System Management Facility (SMF). See also the [logging-related](#) CONFIG control statements.

Logging distributions directly from SMTP-Send

You can also log e-mail distributions directly from SMTP-Send. Log records are written either to an external dataset (batch) or CICS journal (CICS) or using the System Management Facility (SMF). To use any of these features, the feature must first be enabled in the OPSCONFG system configuration file.

To log to a sequential file from a batch job add a LOGFILE DD statement similar to the following to the job's JCL. Note that with sequential dataset logging, OS/390 allows only one SMTP-Send job to use a particular log dataset at a time.

```
//LOGFILE DD DISP=MOD,DSN=logdatasetname
```

To log to a CICS journal or to use SMF, once your system programmer has enabled the feature, no explicit specification is required in batch jobs or CICS transactions.

OP/SS 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.

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.ver.s.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

As you might expect, OP/SS supports all of the SMTP-Send control statements detailed at the beginning of this chapter as well as the configuration statements described in [Customizing](#). 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 and specify the appropriate EXEC parms for the PIPGS3B or RAPIDGS3 program.

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@tbsssoft.com
TO sales@tbs2.com
TO anyone@keane.uk.com
CC support@keane.uk.com
BCC tsol@tbsssoft.com
REPLY-TO tso2@tbsssoft.com

```

IBMMail addressing compatibility

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@tbsssoft.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 use of the additional features provided by SMTP-Send.

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 and send it or you can send an existing data set. 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.

Note: A sample REXX Exec, OPSREXX, is included in INSTLIB to show how to invoke SMTP-Send from REXX. This EXEC does not demonstrate all of the functionality of SMTP-Send but can be used as a starting point to create your own EXEC that will meet your needs.

Composing a message

To compose a message, type the SMTP-Send send CLIST name, normally SMTPSEND, in the ISPF command line and press Enter. You can also compose a message beginning at the SMTP-Send Main Menu. In either case, the following dialog will be presented:

```
----- S M T P - S e n d -----
Command ==>
Enter data and press ENTER to confirm values or END to exit

Send to ...                                     CC BCC
...                                             - -
...                                             - -
...                                             - -
...                                             - -
...                                             - -
...                                             - -
...                                             - -
...                                             - -

Reply-To(optional)..

Subject ...

Reply-To(optional).. potreezeby@tbssoft.com
```

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 appropriate field corresponding to that user. When you exit this panel, an ISPF Edit panel will open. To cancel your send, press Cancel. You can compose your message in the Edit input area as below:

```
File Edit Confirm Menu Utilities Compilers Test Help
EDIT USER.OPS.LISTDSN Compose
Msg
Command ==> Scroll ==>
CSR
***** Top of Data
*****
===== To: foo@bar.com
===== cc: cc@foo.com
===== bcc: bcc@foo.com
===== Reply-To:
===== Subject: Quick note using SMTP
===== -----
-----
===== Enter SEND to send, ADDRESS to revise envelope or END to send later
===== Issue OPSHELP command for usage information
000001 This is the first line of the message
.....
.....
.....
.....
```

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 End. If you want to quit without saving or sending the message, press Cancel.

SMTP-Send Main Menu

Normally you access the SMTP-Send Main Menu by issuing the command OPSMENU from the ISPF command prompt. Your installation may make it available from another panel. The Main Menu panel provides 3 choices as shown below:

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 you compose a message or send a file. 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.

```
----- S M T P - S e n d -----
Command ==>

0 Settings          Change your SMTP-Send defaults
1 Compose a Message Compose a message and send it
2 Send a File       Send a file

User ...: user@foo.com
Userid .: IBMUSER
```

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 Defaults -----
Command ==>
Press ENTER to save these values. Press END to quit
Sender's userid ....: IBMUSER
      e-mail ID .. user@foo.com
Reply-To(optional)... Document type ..... TEXT Lines per page .. 66

SMTP-Send Datasets
CONTROL ..... IBMUSER.SMTPSEND.CONTROL
EDITFILE ..... IBMUSER.SMTPSEND.MESSAGE
CONFIGURATION .. OPS.CONFIG
LOAD ..... OPS.LOADLIB
```

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

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 [distribution control statements](#).

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 yourname.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 yourname.com
INTYPE TEMPLATE XML
OUTTYPE XMLMERGE
//MESSAGE DD * MIME Version: 1.0
... From:
    &company_name;
    &company_address;
...
    &#169; 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:

```


<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

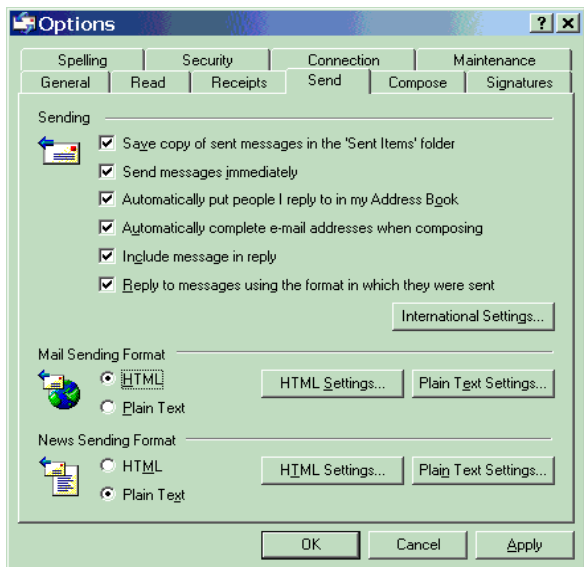
The HTML source file that you created in step 1, will usually reference other files, for instance graphics in GIF or JPG format. In this step, called flattening, you will convert all of the constituent files of the web page into a single text stream. This stream contains all of the data that the recipient's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

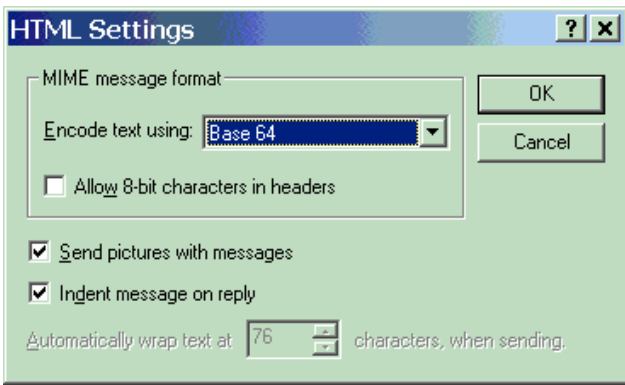
Option settings in Outlook Express determine the format of messages. You must tell Outlook Express to send HTML. In addition, you can create a plain text version of the HTML to allow limited-function e-mail clients like pagers and cellphones to display the message.

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**.

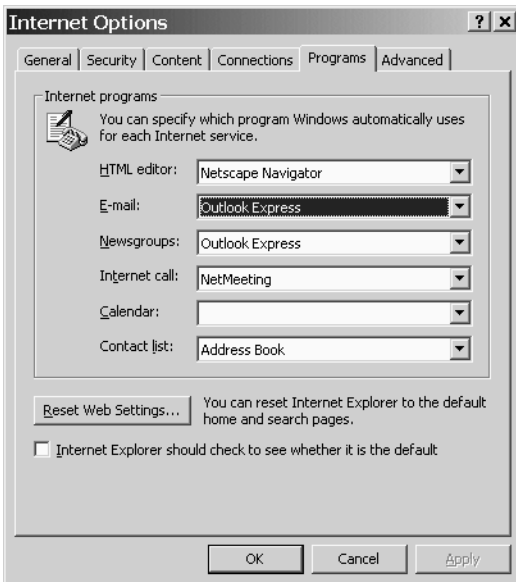


Click **OK** to save your settings, then click **OK** to save your options.

Create alternate format with Microsoft Internet Explorer (IE)

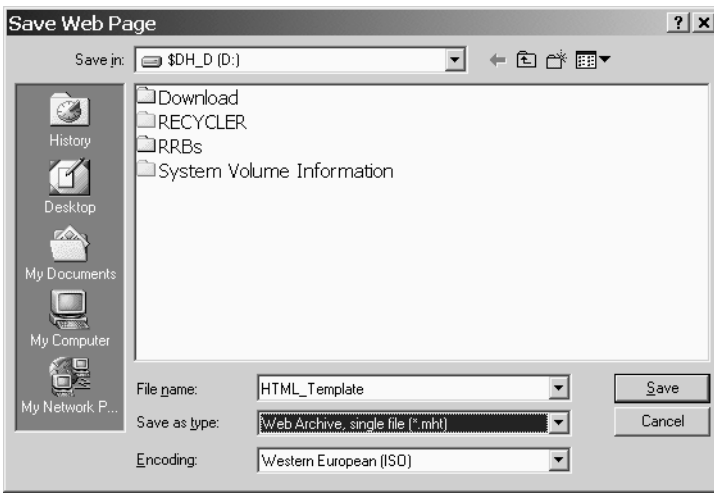
Make sure that Outlook Express is the e-mail client that IE uses.

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 readable. 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.

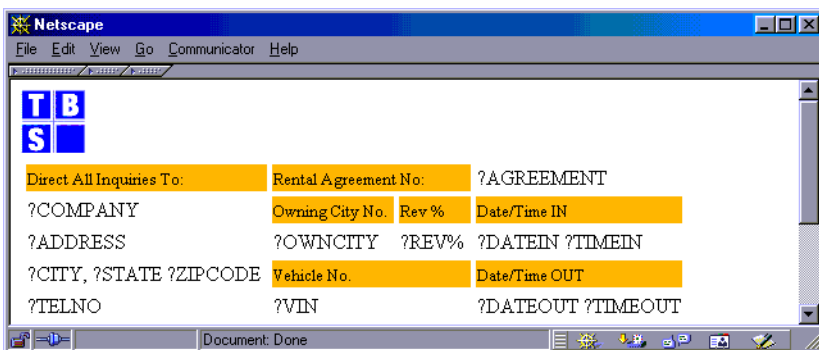
5. Transfer text stream to z/OS or OS/390

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@tbsssoft.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@tbsssoft.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"

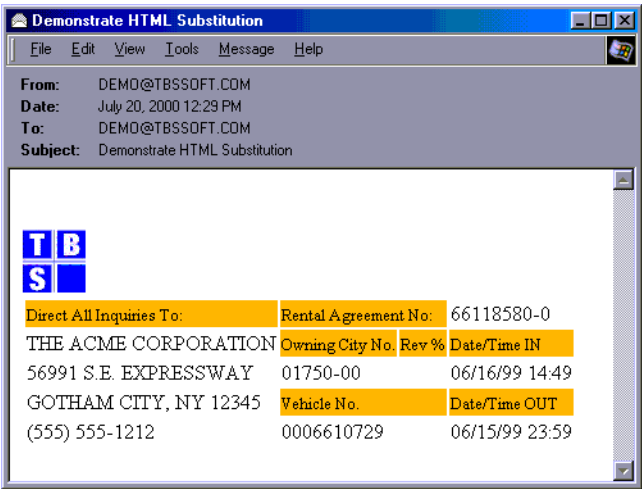
```
<br><font size=3 face="Times New Roman"><br></font>
<!--img src=cid:_1_1AA4000025CC0076AE788525691F-->

<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
R0lGODdhZABkAFcAAAAAAGgAAJkAAMAAAOEAAP8AAABOAGhoAJloAMBoAOFoAP9oAACZAGiZAJmZ
AMCZAOGZAP+ZAADAAGjAAJnAAMDAAOHAAP/AAADhAGjhAJnhAMDhAOHhAP/hAAD/AGj/AJn/AMD/
... THR8zTWQCHbLGz3PNwX4ss4jyilebGNDE/039R8lpfJpeLq+XFtK54zyhif+qv5mIkfbIE9PqD2C
BU7+603j/v09rNy0ZEyjn1bnVoaGABAQA7
--=_related 0076AE7A8525691F_ =--
```

MERGEDATA 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

Installing SMTP-Send

Requirements

- OS/390 (MVS) V1 R3, OS/390 V2 or later or any z/OS release
- IBM TCP/IP V3 R2 or OS/390 V2 Communications Server or later
- under CICS, the TCP/IP socket interface must be active
- **OP/SS** requires at least one of:
 - OfficePath/Batch-TSO
 - OfficePath/CICS
 - RAPID/Batch-TSO
 - RAPID/CICS
- The userid whose credentials are used to run SMTP-Send must be authorized to use OS/390 UNIX System Services (the userid must have an OMVS segment). This is required to allow use of the TCP/IP socket interface utilized by SMTP-Send. Refer to the UNIX System Services Planning documentation for further information.

Distribution package contents

- software on a standard labelled magnetic tape or via FTP
- Installing and Using SMTP-Send (this book)
- a product authorization code (if required)
- a cover letter

Installation data sets

Data Set Number	Data Set Name	Contents
1	OPS.rel.INSTLIB.TAPE	sample installation JCL; sample CICS table entries
2	OPS.rel.LOADLIB.TAPE	load modules
3	OPS.rel.CICSLOAD.TAPE	CICS-specific load modules
4	OPS.rel.ISPPLIB.TAPE	ISPF dialog panels
5	OPS.rel.ISPCLIB.TAPE	ISPF dialog CLISTs
6	OPS.rel.ISPMLIB.TAPE	ISPF dialog messages
7	OPS.rel.DOC.PDF	documentation in PDF format
8	OPS.rel.WIN32.SRCHREP.WIN32.EXE	HTML mail merge PC test program

Authorization codes

Your license to use SMTP-Send comes with a 12-digit authorization code. The authorization code reflects the type of license you have. For more information about how authorization codes work, see the Cover Letter.

If you are going to run SMTP-Send in demonstration mode only, you do not need an authorization code, and you can skip the installation step that supplies this code.

Installing SMTP-Send

Before you start the installation

Use the jobs in the INSTLIB installation data set to install SMTP-Send. Before you submit these jobs for execution:

- change XX to // in all job statements
- add the accounting information required for your system.

Make sure that:

- you read the comments at the beginning of each job
- each job executes with a condition code of zero (00).

When you install SMTP-Send, you can either use the default values that are provided in the installation JCL or change some of the defaults to:

- make SMTP-Send operate in a language other than English
- make SMTP-Send fit the direct access storage device (DASD) you are using
- add parameters to specify accounting information or a message class
- make SMTP-Send conform to the standards and practices of your organization.

Unless configured to communicate directly with an SMTP relay server, SMTP-Send requires either IBM's SMTP Server or the [TBS SMTP Spool Server](#) started task to be active. Refer to the Customization or Administration guide for your release of TCP/IP for information on how to configure and start IBM's SMTP server.

SMTP-Send is written using the SAS/C compiler and requires the availability of the SAS/C runtime environment. All of the required modules are shipped in the SMTP-Send LOADLIB and CICSLOAD data sets. For more information on the version of SAS/C used to create your software, see the SASCVERS member in INSTLIB. Member SASDESC describes the runtime environment expected by SMTP-Send.

To install SMTP-Send

Step Procedure

- 1 Create and run a job to copy the installation JCL data set from the distribution tape. Copy the installation JCL (if you obtained SMTP-Send software on tape) or download the SMTP-Send software.

Use the JCL below as a model when you create this job. You will need to tailor the accounting parameters on the job card to work with your system.

```

XXOPSI00 JOB (accounting),CLASS=A
//*****
//*
//* FUNCTION: UNLOAD THE INSTALLATION LIBRARY FROM TAPE
//* INPUT : SMTP-SEND DISTRIBUTION TAPE
//* OUTPUT : INSTALLATION DATA SET "OPS.rel.INSTLIB"
//*
//* Update the lines indicated by <=== to meet the
//* requirements of your system.
//*
//*****
//UNLOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//TAPEIN DD DSN=OPS.rel.INSTLIB.TAPE,          <===
// UNIT=3480,VOL=SER=OPSnnn,                 <===
// LABEL=(1,SL),DISP=OLD
//DISKOUT DD DSN=OPS.rel.INSTLIB,            <===
// UNIT=3380,VOL=SER=vvvvvv,                 <===
// SPACE=(TRK,(4,1,15),RLSE),
// DISP=(NEW,CATLG,DELETE)
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
COPY INDD=( (TAPEIN,R) ),OUTDD=DISKOUT
/*

```

- 2 Run installation job OPSI02 to copy the SMTP-Send program materials.
- 3 If SMTP-Send will be used under CICS, run installation job OPSI03 to add required CICS file definitions and update program definitions. Add CICSLOAD, then LOADLIB, to your DFHRPL dataset.
- 4 Run installation job OPSI04 to create the system OPSCONFG VSAM data set. If SMTP-Send will be used under CICS and you do not use RDO to dynamically allocate the file, add the VSAM file to your CICS JCL.
- 5 Make sure that the MVS SMTP Server started task is running. Verify your SMTP-Send installation with job OPSI05. Specify your e-mail destination address according to the commented instructions in the JCL. Use the OPST transaction to test your CICS installation.
- 6 **OP/SS** Run installation job OPSI06 to compile and link the name of the system CONFIG file data set in load module OPSCONFIG. If OP/SS will be used under CICS, make sure the CONFIG file data set name is specified for the OPSCONFG CICS file definition via either RDO or JCL.
- 7 **OP/SS** Run installation job OPSI07 to rename your existing OfficePath or RAPID load modules and copy the OP/SS modules into your current OfficePath or RAPID data set. When OPSI07 finishes you can use the PIPT (OfficePath) or ZAPT (RAPID) transaction to verify OP/SS under CICS.
- 8 If you plan to offer SMTP-Send support to your ISPF users, run job OPSI09 to copy the necessary files for the ISPF dialogs. Update the OPSDEFLT clist as required to meet your installation defaults.

OP/SS Be sure to change the OPSSNADS variable to Y if you want to allow users to address messages using SNADS format addresses such as *userid* or *userid.address*.

CICS installation

If you plan to send messages from CICS, make sure that the CICS TCP/IP socket interface is active. 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).)

APF authorization

If you plan to use MVS security services in an SMTP-Send addressing exit or to log distributions to System Management Facility (SMF) from batch or TSO then SMTP-Send must be APF authorized. You must (a) relink the SMTP-Send load module with the AC(1) attribute and (b) authorize the SMTP-Send load library. INSTLIB member, OPSI11 contains JCL to relink the batch load module with the AC(1) attribute. If you will be calling SMTP-Send from TSO and require APF authorization, add OPSGS3BA to the AUTHTSF section of SYS1.PARMLIB member IKJTSO00 and call OPSGS3BA instead of OPSGS3B.

OP/SS For OfficePath or RAPID use PIPGS3BA or ZAPGS3BA respectively.]

It is not possible to APF authorize the SMTP-Send load module under CICS. If you are using RACF (or equivalent) security, then use the CICS-supplied security functions as described in the IBM book CICS-RACF Security Guide.

For more information about APF authorization, see the IBM book MVS/ESA Initialization and Tuning Guide. Remember that all other libraries that are concatenated with the SMTP-Send load library must also be APF-authorized.

Logging distributions

If you plan to write log records either to an external dataset (batch) or journal (CICS) or to use the System Management Facility (SMF), then that feature must be enabled in the OPSCONFG system configuration file.

Logging from batch to SMF

You must ensure that SMTP-Send is APF authorized as described in the previous section.

Logging from batch to a sequential file

If you decide to log to a sequential file then create an empty dataset using JCL similar to the following:

```
//ALLOC EXEC PGM=IEFBR14
//USERLOG DD DSN=logdatasetname,DISP=(NEW,CATLG),
// UNIT=3380,VOL=SER=vvvvvv,SPACE=(CYL,(n,n),
// DCB=(DSORG=PS,RECFM=VB,LRECL=32756)
```

Since this dataset will be used by all batch jobs that invoke SMTP-Send, you should also include a DD statement for it in all SMTP-Send JCL procedures, for example:

```
//LOGFILE DD DSN=logdatasetname,DISP=MOD
```

7

SMTP-Send does not manage the sequential file to prevent x37 abends for out-of-space conditions.

Logging from CICS

If you want to log distributions from CICS you must define an appropriate journal entry in the CICS JCT table. This is a sample for CICS/ESA V4R1:

```
DFH03AB DFHJCT TYPE=ENTRY,      *
                JFILEID=02,     *
                BUFSIZE=32756,   *
                FORMAT=SMF,      *
                JTYPE=SMF
```

Journal definitions may be created and updated online with later versions of CICS.

Use JTYPE=SMF to write to SMF. Use JTYPE=DISK2 to write to user defined JOURNALS 2A & 2B

Soft*Switch Application Toolkit Feature

The Soft*Switch Application Toolkit feature provides a program to replace the SSWDA program provided by SoftSwitch. During initial testing of this feature, it is advisable to create testing JCL based upon your existing JCL and to update only the testing JCL to prevent any interference with your production jobstreams.

If you have never used OfficePath previously

If OfficePath has never been installed by your organization, your distribution materials will be specially packaged to support the distribution of items via the Soft*Switch Toolkit API in the absence of a previous installation of OfficePath. The additional required OfficePath modules will have been already included in the LOADLIB libraries shipped or downloaded.

Testing the Application Toolkit Feature

To test the feature, skip installation step 7, 'Update your existing OfficePath libraries'. Instead add the OPS.Rxx.LOADLIB to your Toolkit API JCL either before or in replacement of, your existing Soft*Switch Toolkit API libraries. In addition, add the SMTP-Send configuration file to your Toolkit API programs by adding the file with DDname OPSCONFG to your JCL. This will insure that the replacement Toolkit API program is used instead of the modules shipped by Soft*Switch.

Installing SMTP-Send in production without JCL changes

To avoid any JCL changes for production usage:

- a) rename the SSWDA program in your Soft*Switch Toolkit API LOADLIB and copy the OP/SS modules into your Soft*Switch Toolkit API LOADLIB or
- b) rename the Soft*Switch Toolkit API and OP/SS LOADLIB datasets so that the OP/SS library replaces your Soft*Switch Toolkit API LOADLIB.

Then run INSTLIB member, OPSI06, to inform OP/SS of the data set name of your SMTP-Send configuration file.

Configuring SMTP-Send

Creating the OPSCONFG file

The OPSCONFG file contains system defaults for SMTP-Send as well as optional distribution list information. The OPSCONFG file is a VSAM KSDS file. This file is created by the OPSMKCFG program using input from two DD statements, CONFIG for general configuration information, DISTLIST to define distribution lists. The VSAM KSDS must be empty for the OPSMKCFG program to successfully store the information. The INSTLIB member, OPSI08, deletes and defines a new cluster before running OPSMKCFG.

OP/SS The OPSCONFG file contains system defaults for OP/SS as well as tables to map your existing 8.8 SNADS-format user addresses to SMTP format. The OPSCONFG file is a VSAM KSDS file. This file is created by the OPSCONFG program using input from three DD statements, CONFIG for general configuration information, USERS to map user names, and NODES to map node names. The VSAM KSDS must be empty for the OPSCONFG program to successfully store the information. The INSTLIB member, OPSI08, deletes and defines a new cluster before running OPSCONFG.

Syntax conventions

Each input file consists of a series of control statements, one statement per line.

Control statements consist of two fields, Field1 and Field2. Field1 must begin in the first column. Field2 begins at the first non-blank position following Field1 and is limited to 256 characters in length.

A control statement that begins with an asterisk (*) is a comment and is not processed. Likewise, blank control statements are ignored.

All Field1 values must be in upper-case. Field2 values are case-insensitive.

To continue a statement, place the first part of Field2 in quotes immediately followed by a plus sign (+). Each continued statement must begin with the same Field1 followed by the next part of Field2.

For example:

```
COMMAND "parm1,"+
COMMAND "'parm value 2'"'+
COMMAND ",parm3"
```

is equivalent to:

```
COMMAND parm1,"parm value 2",parm3
```

Creating the OPSCONFG VSAM data set

```
XXOPSI08 JOB (accounting),CLASS=A
// EXEC PGM=OPSMKCFG,REGION=7192K
//STEPLIB DD DSN=OPS.ver.s.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CONFIG DD DSN=OPS.ver.s.INSTLIB(CONFIG),DISP=SHR
//DISTLIST DD DSN=OPS.ver.s.INSTLIB(DISTLSTS),DISP=SHR
//OPSCONFG DD DSN=OPS.ver.s.OPSCONFG,DISP=SHR
//*
```

Typically you would create one OPSCONFG file for production usage and one or more variants for testing.

CONFIG control statements

Specifying the default domain name

The first control statement specifies the SMTP domain name for your enterprise. This domain name will be appended to all sender and recipient name that do not specify a domain name. You must include a DEFAULT_DOMAIN control statement.

DEFAULT_DOMAIN

DEFAULT_HOST

Purpose: specifies the default SMTP domain name for your enterprise

OP/SS DEFAULT_HOST serves a similar purpose to the OfficePath or RAPID "DEFAULT NODE" commands.

Syntax: DEFAULT_DOMAIN domain

Example: DEFAULT_DOMAIN FOO.COM

Setting the authorization code

SMTP-Send requires a 12-digit product authorization code that reflects each optional product feature that you have licensed. You can set the authorization code values using the following control statements:

BATCH_AUTHCODE

Purpose: specifies the SMTP-Send Batch feature authorization code
Syntax: `BATCH_AUTHCODE nnnn,nnnn,nnnn`
Default: none, SMTP-Send Batch operates in "demonstration" mode
Example: `BATCH_AUTHCODE 1234,5678,4321`

CICS_AUTHCODE

Purpose: specifies the SMTP-Send CICS feature authorization code
Syntax: `CICS_AUTHCODE nnnn,nnnn,nnnn`
Default: none, SMTP-Send CICS operates in "demonstration" mode
Example: `CICS_AUTHCODE 1234,5678,4321`

HTML_AUTHCODE

Purpose: specifies the SMTP-Send HTML Substitution feature authorization code
Syntax: `HTML_AUTHCODE nnnn,nnnn,nnnn`
Default: none, SMTP-Send HTML operates in "demonstration" mode
Example: `HTML_AUTHCODE 1234,5678,4321`

For more information on authorization codes, see the cover letter that came with your SMTP-Send distribution package.

Specifying the SMTP Server timezone offset

All times passed to the SMTP server must specify the offset of the host system from Universal Time (also known as GMT). The default value is "-500" or Eastern Standard time. If your location observes Daylight Savings Time, be sure to specify the offset for Standard time. SMTP-Send will automatically adjust for Daylight Savings Time.

TIMEZONE

Purpose: specifies the offset from UT of your host system from which you are sending items
Syntax: `TIMEZONE {+|-}HHMM`
Default: `TIMEZONE -0500`
Example: `TIMEZONE +0100`

Specifying observance of Daylight Savings Time

To correctly create the time zone offset for times passed to the SMTP server, SMTP-Send needs to know if the location of your SMTP server observes Daylight Savings Time.

DST

Purpose: specifies the offset from SMTP domain name of the host system from which you are sending items
Syntax: `DST Yes|No`
Default: `DST Yes`
Example: `DST Yes`

Specifying Daylight Savings Time rules

When Daylight Savings Time (DST) is being observed, SMTP-Send needs to know the rules for determining when a date/time value is within the DST period or outside of that period. The rules are specified by supplying the month, week and day of the week when DST begins and ends.

By default, the DST period is defined as the second Sunday in March to the first Sunday in November. (This is the period during which DST is observed in the United States.) The variables *begin_month*, *begin_week*, and *begin_day*, define the month, week, and day, respectively, when DST begins. The variables *end_month*, *end_week*, and *end_day* define when DST ends. Months are 1 (January) through 12 (December); weeks are 1 through 4; days are 1 (Sunday) through 7 (Saturday). You can use negative numbers to specify the weeks, where -1 is the last week of the month, -2 is the second to last week, and so on.

DSTLAW

Purpose: specifies the period for Daylight Savings Time
Syntax: `DSTLAW "begin_month begin_week begin_day end_month end_week end_day"`
Default: `DSTLAW "3 2 1 11 1 1"`
Example: `DSTLAW "4 -2 1 10 -1 1"`

Changing the SMTP Server information

By default, SMTP-Send sends text message items by writing output to a SYSOUT external writer on SPOOL. The default external writer name is "SMTP" and the default output class is "B." When running under CICS, the JES node name is also required, the default used is "N1."

Binary items such as PC files, HTML, RFT-DCA are sent using MIME-encodings directly to an SMTP relay server using TCP/IP as the transport mechanism. The default SMTP Server IP address is built by appending the default domain to the string "SMTP." The default TCP/IP port number for SMTP is 25. You can change the port number used by appending the ':' character followed by the desired number to the address value.

You can change these values using the following control statements:

ATSIGN

Purpose: identifies the character that separates the local part of an e-mail address from the domain portion. The ATSIGN configuration statement was introduced by IBM APAR PQ36249. Specify the same value as in your SMTP Server configuration data set (DDname CONFIG in the SMTP started task).

If ATSIGN is not specified in your SMTP Server configuration data set then you do not need to specify it for SMTP-Send.

Syntax: ATSIGN *char*

Default: ATSIGN @

Example: ATSIGN #

SetTCPFailureRC

Purpose: specifies the return code to be returned when SMTP-Send is unable to connect to both the primary and secondary SMTP server to send a message via TCP/IP.

Syntax: SetTCPFailureRC *retcode*

Default: SetTCPFailureRC 4

Example: SetTCPFailureRC 8

SMTP_RELAY_ADDRESS

Purpose: specifies the primary SMTP relay server that is used to send e-mail via TCP/IP sockets.

Syntax: SMTP_RELAY_ADDRESS {*Server|IPaddress*][:*port*]

Default: SMTP_RELAY_ADDRESS 127.0.0.1:25

Example: SMTP_RELAY_ADDRESS smtp.foo.com:725

SMTP_RELAY_ADDRESS_ALTERNATE

Purpose: specifies the alternate SMTP relay server if a connection to the primary server cannot be established.

Syntax: SMTP_RELAY_ADDRESS_ALTERNATE {*Server|IPaddress*][:*port*]

SMTP_SERVER_CLASS

Purpose: specifies the MVS SMTP server external writer class.

Syntax: SMTP_SERVER_CLASS *classchar*

Default: SMTP_SERVER_CLASS B

Example: SMTP_SERVER_CLASS X

SMTP_SERVER_NAME

Purpose: specifies the MVS SMTP server external writer name. Use the name of the SMTP started task.

Syntax: SMTP_SERVER_NAME *name*

Default: SMTP_SERVER_NAME SMTP

Example: SMTP_SERVER_NAME TBSSMTP

SMTP_SERVER_NODE

Purpose: specifies the JES node that the MVS SMTP server runs on.

Syntax: SMTP_SERVER_NODE *name*

Default: SMTP_SERVER_NODE N1

Example: SMTP_SERVER_NODE N2

WTO_onTCPFailure

Purpose: specifies that SMTP-Send should issue a WTO when unable to connect to both the primary and secondary SMTP server to send a message via TCP/IP. The text of the message 'OPS126E FAILURE RELAYING TCP/IP DATA' is defined in the SMTP-Send message table OPSSTRTO as string number 126.

Syntax: WTO_onTCPFailure {No|Yes}

Default: WTO_onTCPFailure No

Example: WTO_onTCPFailure Yes

Controlling the SMTP transmission

By default, SMTP-Send sends messages in the MIME format using TCPIP as the transport mechanism. Attachments may be sent as binary data or translated from EBCDIC to ASCII.

You can change this behavior using the following control statements:

DEFAULT_ATTACHMENT

Purpose: defines the formatting for text attachments. Binary attachments are not affected by this setting. To override this setting for a specific SEND operation, specify OUTTYPE RTF or ATTACH attachment TRANSLATE RTF

Syntax: DEFAULT_ATTACHMENT {RTF|NORTF}

Default: DEFAULT_ATTACHMENT NORTF

Example: DEFAULT_ATTACHMENT RTF

MAXMAILBYTES

Purpose: limits the size of e-mails that are passed to the outbound SMTP server. The value should be the same as specified in your SMTP server configuration. For the MVS SMTP server this value is specified via the MAXMAILBYTES statement of the SMTP Server Configuration data set (DDname CONFIG in the SMTP started task). Refer to the documentation for your SMTP server if you use a different server for TCP/IP connections. The minimum value is 1 and the maximum value is 2,147,483,647.

Syntax: MAXMAILBYTES nnnnnnnnnn|nnnnnnnnnK|nnnnnnnnnM

Default: MAXMAILBYTES 512K

Example: MAXMAILBYTES 10M

RC8_on_NoData

Purpose: Overrides the default return code (RC=4) returned when the input MESSAGE dataset is empty and no attachments are requested or all attachments are empty. An empty MESSAGE dataset will not result in a NoData condition if one or more attachments contain data.

Syntax: RC8_on_NoData {Yes|No}

Default: RC8_on_NoData No

Example: RC8_on_NoData Yes

SEND_TEXT

Purpose: indicates the preferred route for simple outbound SMTP messages. Specifying TCPIP results in delivery of e-mail via a direct TCP/IP socket interface to an SMTP server capable of relaying the message. If unable to contact both the primary and secondary SMTP server via TCP/IP, SMTP-Send continue as if SPOOL was specified for that distribution only. Specifying SPOOL results in delivery of e-mail to an MVS-based SMTP relay server via the JES SPOOL interface. The IBM-supplied SMTP Server started task or the SMTP-Send Spool Server are both examples of supported SMTP relay servers.

Syntax: SEND_TEXT {TCPIP|SPOOL}

Default: SEND_TEXT TCPIP

Example: SEND_TEXT SPOOL

Highlighting Notices in Microsoft Outlook

You can highlight Notices that are received by users of Microsoft Outlook. This feature works as if the sender had used Outlook's Flag for Follow Up option in the Actions menu. This feature may also work with other e-mail client software that recognizes the X-Message-Flag parameter in the SMTP header.

MSG_IN_XMESSAGEF

Purpose: specifies that Notices are highlighted using Outlook's Flag for Follow Up feature.

Syntax: MSG_in_XMESSAGEF [Yes|No]

Default: MSG_in_XMESSAGEF No

Example: MSG_in_XMESSAGEF Yes

Redirecting output

Normally you would want SMTP-Send to send items to recipients using your SMTP server, however you can optionally redirect the transmission. When running in Batch, redirection is to a sequential file. When running under CICS, redirection is to a CICS Temporary Storage Queue. This feature is useful when you want to test an e-mail enabled application without actually sending items to recipients. Redirection can be specified for both Batch and CICS in the same CONFIG file.

TESTOUTDSN

Purpose: redirects batch output to a sequential data set

Syntax: TESTOUTDSN datasetname[,keywords...]

Example: TESTOUTDSN TBS123.OPSMTM.OUTPUT,DISP=SHR

SMTP-Send always supplies DDNAME, DSORG, RECFM, LRECL and BLKSIZE attributes. The value of DDNAME is DATAOUT. If required you can supply additional attributes using standard MVS JCL keywords or through dynamic allocation. See "Dynamic allocation keywords" for a list of valid keywords.

TESTOUTTSQ

Purpose: redirects CICS output to a CICS Temporary Storage Queue
Syntax: TESTOUTTSQ *8-character-tsqueue*
Example: TESTOUTTSQ PIPTSQO

Note: CICS Temporary Storage queue names are case sensitive. Normally you would not include TESTOUTDSN or TESTOUTTSQ in your production OPSCONFG file. Instead create a special OPSCONFG file for running tests that contains a TESTOUTDSN or TESTOUTTSQ statement.

Handling addressing errors

If an addressing error occurs for one or more recipients, SMTP-Send sends the message to the remaining recipients. SMTP-Send automatically eliminates some forms of addressing errors by append your domain name to an address without one. You can turn this behavior off and create additional addressing errors with the `Append_Domain_Name` statement

If you want to stop SMTP-Send when it encounters an addressing error, set `FAIL_ON_ERROR` to YES. Set `MSG_ON_ERROR` to YES to return the message to the sender's REPLY-TO address.

When `FAIL_ON_ERROR` is set to NO, SMTP-Send sends the message as long as the sender and one or more recipient addresses are valid. If there are no valid addresses, SMTP-Send ends with a non-zero return code.

To enhance security, you can also instruct SMTP-Send to terminate with a return code of 32 when an address is defined by your address validation exit by setting `Fail_on_Validation_Error` to YES

OP/SS To maintain compatibility with the behavior of the original OfficePath and RAPID software, even when OP/SS is unable to map or locate at least one recipient address, it ends execution with a condition code of zero. You can override this characteristic using the `ADDRESS_MAP_FAIL_OK` statement.

APPEND_DOMAIN_NAME

Purpose: controls whether or not the default domain name is added to a TO, CC or BCC recipient value when no domain has been specified.
Syntax: APPEND_DOMAIN_NAME {Yes|No}
Default: APPEND_DOMAIN_NAME Yes
Example: APPEND_DOMAIN_NAME No

FAIL_ON_ERROR

Purpose: controls the SMTP-Send actions when addressing errors are encountered
Syntax: FAIL_ON_ERROR {Yes|No}
Default: FAIL_ON_ERROR No
Example: FAIL_ON_ERROR Yes

FAIL_ON_VALIDATION_ERROR

Purpose: controls the SMTP-Send actions when addresses fail validation
Syntax: FAIL_ON_VALIDATION_ERROR {Yes|No}
Default: FAIL_ON_VALIDATION_ERROR No
Example: FAIL_ON_VALIDATION_ERROR Yes

MSG_ON_ERROR

Purpose: controls the sending of messages when addressing errors are encountered
Syntax: MSG_ON_ERROR {Yes|No}
Default: MSG_ON_ERROR No
Example: MSG_ON_ERROR Yes

ERROR_MSG_TO_NAME

Purpose: specifies the SMTP address to which error messages will be sent if a reply address cannot be obtained from either a REPLY-TO or FROM statement. If no domain is specified, the `default_domain` value will be appended.
Syntax: ERROR_MSG_TO_NAME *name*
Default: ERROR_MSG_TO_NAME postmaster@default_domain
Example: ERROR_MSG_TO_NAME postmaster@foo.com

ERROR_MSG_FROM_NAME

Purpose: specifies the SMTP address from which error messages will be sent. If no domain is specified the `default_domain` value will be appended.
Syntax: ERROR_MSG_FROM_NAME *name*
Default: ERROR_MSG_FROM_NAME OPSend@default_domain
Example: ERROR_MSG_FROM_NAME batch [or batch@tbsssoft.com]

EXPAND_DIST_LIST

Purpose: controls the expansion of distribution lists in the message headers. Default value `YES` includes the e-mail address of each member of a distribution list in the message headers. Value `NO` suppresses this information.

Syntax: `EXPAND_DIST_LIST {Yes|No}`

Default: `EXPAND_DIST_LIST Yes`

Example: `EXPAND_DIST_LIST No`

UseDLAddress

Purpose: controls the expansion of Distribution lists during LDAP processing. By default, SMTP-Send will not expand a Distribution List with an email address associated with it, but instead will merely send the email to that address, letting the receiving email system expand the address. This is a change from the default behavior in releases prior to Release 4.2.13 which can be restored by specifying `UseDLAddress No`

Syntax: `UseDLAddress {Yes|No}`

Default: `UseDLAddress Yes`

Example: `UseDLAddress No`

Enabling user exits

You can further customize SMTP-Send by writing user exits. The following statements tell SMTP-Send which exits you want to use. For information on how to code these exits, see [Customizing user exits](#).

VALIDATEPGM

Purpose: specifies a program to validate addresses and optionally to reformat messages for particular addresses such as pagers. The program name is limited to 7 characters as SMTP-Send appends the character 'B' or 'C' to the name when running in Batch/TSO or CICS respectively.

Syntax: `VALIDATEPGM program[,variables]`

Default: `none`

Example: `VALIDATEPGM VALPROG`

DISTLISTPGM

Purpose: specifies a program to expand distribution lists. The program name is limited to 7 characters as SMTP-Send appends the character 'B' or 'C' to the name when running in Batch/TSO or CICS respectively.

Syntax: see: [Using LDAP](#)

LDAP_LOOKUP_PGM

Purpose: specifies a program to look up e-mail addresses. The program name is limited to 6 characters as SMTP-Send may append additional characters such as 'B' or 'C' to the name when running in Batch/TSO or CICS respectively.

Syntax: see: [Using LDAP](#)

LDAP_DISTLIST_PGM

Purpose: specifies a program to expand e-mail distribution lists. The program name is limited to 6 characters as SMTP-Send may append additional characters such as 'B' or 'C' to the name when running in Batch/TSO or CICS respectively.

Syntax: see: [Using LDAP](#)

DATAEXITPGM

Purpose: specifies a program to manipulate the message data before it is passed to SMTP for distribution. The program name is limited to 7 characters as SMTP-Send appends the character 'B' or 'C' to the name when running in Batch/TSO or CICS respectively. This definition does not cause the exit program to be invoked. You must request the data exit be invoked either through the `ALWAYSALLDATAEXIT` configuration statement or from the address validation exit.

Syntax: `DATAEXITPGM program[,variables]`

Default: `none`

Example: `DATAEXITPGM DATAXT`

ALWAYSALLDATAEXIT

Purpose: specifies whether the data manipulation exit is invoked for each SMTP-Send request. The invocation of the data exit can also be controlled from the address validation exit.

Syntax: `ALWAYSALLDATAEXIT {Yes|No}`

Default: `ALWAYSALLDATAEXIT No`

Example: `ALWAYSALLDATAEXIT Yes`

EXIT_PASSWORD

Purpose: Provides the password used in the MAPPINGPGM exit. This value is encrypted in the OPSCONFG VSAM file. For the DISTLISTPGM exit program, the LDAP_Password value is used when the exit program name begins with LDAP and the Exit_Password value is used otherwise.

Syntax: Exit_Password *value*
Default: none

LDAP_PASSWORD

Purpose: Provides the password value to be encrypted in the OPSCONFG VSAM file. Specify a * in actual LDAP statements where the password is to be replaced

Syntax: LDAP_Password *value*
Default: none

LDAP_OPENRETRY

Purpose: number of times to attempt to connect to LDAP server

Syntax: LDAP_OpenRetry *count*
Default: LDAP_OpenRetry 3

LDAP_OPENDELAY

Purpose: delay between connection attempts to LDAP server

Syntax: LDAP_OpenDelay *seconds*
Default: LDAP_OpenRetry 5

LDAP_TIMEOUT

Purpose: time to wait before aborting LDAP search

Syntax: LDAP_Timeout *seconds*
Default: LDAP_Timeout 60

Controlling the look of the message

CREATE_MAILTO_LINKS

Purpose: controls whether or not email addresses and URLs are converted into HTML links when encountered in the subject or the body of a text message.

Syntax: Create_MailTo_Links {*Y|N*}
Default: Create_MailTo_Links Y

BODY_FONT-SIZE

Purpose: specifies the font size used in the Cascading Style Sheet specifications for the body of the text. The receiving email client will use the first value it recognizes.

There are two types of font-family values:

font-family: The name of a font-family, like "verdana" or "arial"

generic-family: The name of a generic font-family, like "serif" or "sans-serif"

Syntax: BODY_Font-Size {*pointsize* |
{*xx-small|x-small|small|medium|large|x-large|x-large|xx-large*}}

Default: BODY_Font-Size medium

Example: BODY_Font-Size 10px

BODY_FONT-FAMILY

Purpose: specifies the Font family used in the Cascading Style Sheet specifications for the body of the text.

Syntax: BODY_Font-Family {*font [,font...]*}

Default: BODY_Font-Family none

Example: BODY_Font-Family sans-serif

BODY_COLOR

Purpose: specifies that color used for the foreground and background of the body of the text.

Syntax: BODY_Color {*background_color[,foreground_color]*}

where color choices are:

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

Default: BODY_Color {White,Black}

HTMLFONT

Purpose: defines the relative font size of preformatted text to be defined using the HTML <FONT=value> tag. The default value of none,

indicates that no HTML tags are to be included with the text. HTMLFONT is ignored if the data is not being converted from plain text to HTML

Syntax: HTMLFONT { |1|2|3|4|5|6|7 }

Default: HTMLFONT none

HTMLTEXTHEADER

Purpose: specifies HTMLtag used to indicate the start of preformatted text when 1403/text data is being converted to HTML.

Syntax: HTMLTextHeader *text*

Default: HTMLTextHeader <PRE>

Example: HTMLTextHeader "<font=Helv SIZE=2><PRE>

HTMLTEXTTRAILER

Purpose: specifies HTMLtag used to indicate the end of preformatted text when 1403/text data is being converted to HTML

Syntax: HTMLTextTrailer *text*

Default: HTMLTextTrailer </PRE>

Example: HTMLTextTrailer "</PRE>

ORIENT

Purpose: specifies the default orientation used for RTF attachments when no value is specified in the RTF subparameter of OUTTYPE

Syntax: ORIENT {Portrait | Landscape}

Default: ORIENT Landscape

PITCH

Purpose: specifies the default point size used for RTF attachments when no value is specified in the RTF subparameter of OUTTYPE or a PITCH control statement

Syntax: PITCH *pointsize*

Default: PITCH 12

REMOVETRAILINGBLANKS

Purpose: controls whether or not SMTP-Send should remove trailing blanks from each line of data before sending

Syntax: RemoveTrailingBlanks {Yes|No}

Default: RemoveTrailingBlanks Y

REPLACELFWCRLF

Purpose: specifies whether LineFeed control characters are converted to CarriageReturn LineFeed control sequences in text HFS files.

Syntax: ReplaceLfwCRLF {Y|N}

Default: ReplaceLfwCRLF Y

SENSITIVITY_MSG

Purpose: location of the sensitivity text. The actual text is controlled through the OPSCONFG statements SENSITIVITY_TXT

Syntax: Sensitivity_Msg {None | Top | Bottom}

Default: Sensitivity_Msg None

SENSITIVITY_TXT

Purpose: code as many SENSITIVITY_TXT statements as required for your text. HTML formatting tags may be used in the text. See the example in INSTLIB(CONFIG).

Personal Sensitivity (Type=1)

Private Sensitivity (Type=2)

Confidential Sensitivity (Type=3)

Company Confidential Sensitivity (Type=4)

Syntax: Sensitivity_Txt *type,text*

Default: Sensitivity_Txt None

Example: Sensitivity_Txt 1, ''

Sensitivity_Txt 1, "This E-mail is Personal \n\"

Sensitivity_Txt 1, "only for the use of the recipient(s)\n"

Sensitivity_Txt 1, ''

USE_CSS_STYLES

Purpose: specifies that HTML messages are formatted using Cascading Style Sheets. The BODY_xxxxx statements control statements control the style characteristics.

A useful primer for CSS usage can be found at <http://www.w3.org/Style/Examples/011/firstcss>

Syntax: Use_CSS_Styles {Y|N}

Default: Use_CSS_Styles Y

USECUSTOMCODEPAGE

Purpose: controls whether or not SMTP-Send should attempt to load an external character translation table, named XLT nnn where nnn is the value specified on a CODEPAGE control statement. The XLT nnn module is created using INSTLIB member OPSI12

Syntax: UseCustomCodepage {Y|N}

Default: UseCustomCodepage N

Controlling conversion to Adobe PDF format

PDFBIAS

Purpose: Screen causes SMTP-Send to create PDF documents suitable for screen display. Text is rendered in 12pt Courier, with page length and width calculated from the lines per page and characters per line, Top, bottom, left and right margins are all set to zero.
Print causes SMTP-Send to use the PDF defaults as below or their overridden values.

Syntax: PDFBIAS {Screen | Print}

Default: PDFBIAS Screen

PDFPITCH

Purpose: specifies the font size to use for text in the PDF document

Syntax: PDFPITCH *pointsize*

Default: PDFPITCH 12

PDFORIENT

Purpose: specifies the orientation to use for text in the PDF document

Syntax: PDFORIENT {Portrait | Landscape}

Default: PDFORIENT Landscape

PDFPAPER

Purpose: specifies common page sizes (height and width may be overridden by PDFWIDTH and PDFHEIGHT statements)

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

Default: PDFPAPER LETTER

PDFWIDTH

Purpose: specifies the page width in inches (the default) or millimeters

Syntax: PDFWIDTH inches | millimetersmm

Default: PDFWIDTH 8.5

PDFHEIGHT

Purpose: specifies the page height in inches (the default) or millimeters

Syntax: PDFHEIGHT inches | millimetersmm

Default: PDFHEIGHT 11

PDFTOPM

Purpose: specifies the top margin in inches (the default) or millimeters

Syntax: PDFTOPM inches | millimetersmm

Default: PDFTOPM 8.5

PDFBOTM

Purpose: specifies the bottom margin in inches (the default) or millimeters

Syntax: PDFBOTM inches | millimetersmm

Default: PDFBOTM 8.5

PDFLM

Purpose: specifies the left margin in inches (the default) or millimeters

Syntax: PDFLM inches | millimetersmm

Default: PDFLM 8.5

PDFRM

Purpose: specifies the right margin in inches (the default) or millimeters

Syntax: PDFRM inches | millimetersmm

Default: PDFRM 8.5

Notes for PDF configuration statements:

leading zeros are required for decimal inch values less than 1. i.e PDFTOPM 0.5 is valid and PDFTOPM .5 is not.
there is no space between the millimeters value and the mm indicator.

i.e. Specify PDFWIDTH 216mm

Debugging SMTP-Send operation

DEBUG

Purpose: shows each sender or recipient address (a) as coded in CONTROL control statements, (b) as altered by an optional customer address mapping exit, if one exists.

Syntax: DEBUG Yes|No|Verbose

Default: DEBUG No

Example: DEBUG Yes

Note: to enable DEBUG only for a particular job, add PARM='=SETDEBUG=YES' on the EXEC PGM=OPSGS3B JCL statement

Example

Here is an example of the CONFIG section of an OPSCONFG file that shows how the above control statements are coded:

```
DEFAULT_DOMAIN      TBSSOFT.COM
AUTHCODE            1234,5678,1234
TIMEZONE            -0500
DST                 yes
DSTLAW              "4 -2 1 10 -1 1"
SMTP_SERVER_CLASS   X
SMTP_SERVER_NAME    TBSSMTP
SMTP_SERVER_NODE    N5
SMTP_RELAY_ADDRESS  123.210.123.210:25
ATSIGN              #
TESTOUTDSN          OPS.TESTOUT,disp=old,ndisp=catlg
TESTOUTTSQ          OPSTSQO
FAIL_ON_ERROR       Yes
ERROR_MSG_FROM_NAME Postmaster@tbsssoft.com
ERROR_MSG_TO_NAME   Errors_go_here@tbsssoft.com
MSG_ON_ERROR        yes
EXPAND_DIST_LIST    No
VALIDATEPGM         VALIDP
DATAEXITPGM         DATAXT
ALWAYSALLDATAEXIT  Yes
DEBUG               VERBOSE
LDAP_Password       LDAPPW
LDAP_OpenRetry      3
LDAP_OpenRetry      5
LDAP_Timeout        60
Exit_Password       "Exit Pw"
Sensitivity_Msg     Bottom
Sensitivity_Msg     None
ReplaceLFwCRLF     Y
Use_CSS_Styles      Y
BODY_Font-Size      medium
BODY_Font-Family    Helv
BODY_Color          {Black,White}
HTMLFONT            5
HTMLTextHeader      "<font=Helv SIZE=2><PRE>"
HTMLTextTrailer     "</PRE></FONT>"
PDFBIAS             Print
PDFPITCH            12
PDFORIENT           Landscape
PDFWIDTH            9
PDFHEIGHT           11
PDFTOPM             0.5
PDFBOTM             0.25
PDFLM               25mm
PDFRM               25mm
```

Adding a standard Notice to each message

You can add a notice at the bottom of each message. This notice can be standard "boilerplate" text that appears in all messages or you can define it explicitly for each message using SMTP-Send control statements.

Use the optional identifier to specify additional text for language-specific code pages or content-specific requirements. To add language-specific text the identifier must be the codepage value. To add context-specific text the identifier can be any 3-character number. If all NOTICE statements contain an identifier, the notice lowest in collating sequence becomes the default notice.

Notice text can be any length, but the text in each NOTICE statement must be no more than 128 characters.

Use the string "\n" to insert a line break. You can also use standard HTML tags.

The notice text that SMTP-Send includes is determined according to the following hierarchy:

1. text in SMTP-Send control statements
2. text in a language-specific codepage
3. text in CONFIG control statements

Notice text is not automatically included in [HTML mail merge](#) template merges. If you want to include notice text, use the keyword symbol ?SS_NOTICE or XML entity &SS_NOTICE; to indicate where to insert the notice text in the template.

NOTICE

Purpose: adds a standard notice at the bottom of each message.

Syntax: NOTICE [nnn,]"text"

Example: NOTICE 037, '
'
NOTICE 037, "The information contained in this e-mail is "
NOTICE 037, "strictly confidential.\n"
NOTICE 037, "If you received this e-mail in error, please "
NOTICE 037, "notify the sender by reply e-mail."
NOTICE 037, '</BR>'

Distribution list control statements

SMTP-Send allows you to group recipients together into distribution lists. These lists can be managed by SMTP-Send or by an external directory manager.

Distribution list control statements are specified in the OPSCONFG file via the DISTLIST DDname. They consist of two components, the distribution list name and the list member. The list name can be up to 18 characters in length and begins in column 1. The list member value can be up to 60 characters in length and does not need to contain the domain name, which will be automatically appended by SMTP-Send if not present.

Repeat the list name component on subsequent control statements to specify additional members of the distribution list. All members of a distribution list must be specified sequentially, one after the other.

Here is an example of the DISTLIST section of an OPSCONFG file that shows a couple of sample distribution lists:

```
MANAGERS      alberta@foo.com
MANAGERS      bernie@foo.com
MANAGERS      charley@foo.com
MANAGERS      denise@foo.com
MANAGERS      emily@foo.com
VENDORS       support@tbsssoft.com
VENDORS       sales@tbsssoft.com
```

When you specify a recipient address that has no domain name, for example, sales, SMTP-Send first checks to see if it is a distribution list name. If it is not a distribution list then SMTP-Send appends the default domain name to create a complete address, for instance, sales@tbsssoft.com.

OP/SS OfficePath/SNADS and RAPID RECIPIENT statements are eligible for distribution list processing only if a single UserID value and no Address value is specified on the statement.

Logging distributions

If you decide to create a log of e-mail distributions from SMTP-Send you must specify where and how to write log records. See also: [Logging distributions](#) in Sending.

LOGGING

Purpose: indicates if SMTP-Send generates log records, and if so, specifies where to write them.

YES - in batch, writes log records to a sequential dataset as defined by the LOGFILE DD statement in the SMTP-Send JCL. Under CICS, writes log records either to a user-defined journal or to SMF depending on the LOG_JNUM specification. SMF - in batch, writes log records to SMF. Under CICS, SMF writes log records in SMF format to a CICS journal based on the LOG_JNUM specification.

Syntax: LOGGING [No|Yes|SMF]

Default: LOGGING No

Example: LOGGING SMF

LOG_JNUM

Purpose: (CICS only) indicates which CICS journal to write log records. The journal number must be a value between 2 and 99. To write data to CICS SMF type 110 records use LOG_JNUM=SMF. In this case the CICS journal definition must be JTYPE=SMF.

Syntax: LOG_JNUM [nn|SMF]

Default: LOG_JNUM 2

Example: LOG_JNUM SMF

SMF_TYPE

Purpose: specifies the SMF record type (or number) that contains log data. In batch, this is a value between 128 and 255. Under CICS, this is defined according to the JTYPEID parameter of the WRITE JOURNALNUM command.

Syntax: SMF_TYPE nnn

Default: SMF_TYPE 128

Example: SMF_TYPE 255

Note: In order to write log data to SMF from batch, SMTP-Send must be APF authorized. Refer to the APF Authorization section in [Installing SMTP-Send](#)

OP/SS NODES and USERS address mapping control statements

OP/SS uses address mapping control statements to specify how 8.8 SNADS-format addresses should be mapped to their SMTP-format (RFC 822) counterparts. SNADS format addresses consist of two components, each up to 8 characters, separated by a period. The first component specifies the name of the message recipient or user, and the second component specifies the host address or node. These are the same values that appear on OfficePath or RAPID RECIPIENT or USERNAME control statements.

SMTP format addresses have the form user@domain.com. For example TBS Software's technical support address is support@tbsssoft.com.

The OPSCONFG file has two address mapping tables. The NODES table maps SNADS node names to SMTP domain names and the USERS table maps SNADS user names to SMTP user names.

OP/SS searches the mapping tables for Field1 values that match SNADS user and node values. When OP/SS finds a match, it uses the value of Field2 to determine the corresponding SMTP value. If OP/SS cannot find a match for the node it substitutes the value specified by DEFAULT_HOST. Finally OP/SS concatenates the user and domain values of with the standard "@" separator.

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

```
USER NAME TSO
RECIPIENT USER2 LOCATN2
RECIPIENT TOUSER PIPHOST1
```

and your OPSCONFG file includes these statements:

```
CONFIG DD:
DEFAULT_HOST tbsssoft.com
```

```
NODES DD:
PIPHOST1 keane.uk.com
LOCATN2 tbs2.com
```

```
USERS DD:
USER2 sales
TOUSER support
TSO tso
```

then the message item will be sent using this header information:

```
FROM: tso@tbsssoft.com
TO: sales@tbs2.com
TO: support@keane.uk.com
```

OP/SS Configuring and customizing Soft*Switch Application Toolkit Feature

The Soft*Switch Application Toolkit feature program, SSWDA can be configured to change the default behavior when certain types of data are being sent. Some customizations are performed using the OPSCONFG control statements and others are performed by altering the SSWDA program itself.

SSWDA OPSCONFG control statements

SSWMSGEncode

Purpose: specifies the manner in which text from the OBJECT dataset is to be displayed when the MSG= statement is specified in the //COMMAND statements. No specifies that OBJECT text be sent instream. Yes specifies that OBJECT text be sent as an attachment . Both specifies that the OBJECT text be included both instream and as an attachment.

Syntax: SSWMSGEncode {No|Yes|Both}

Default: SSWMSGEncode No

Example: SSWMSGEncode Both

SSW_Header_Compatibility

Purpose: specifies the header text inserted before an attachment. Yes instructs SSWDA to
a) convert the DOCUMENTNAME value into a MS-DOS compatible where the 1st eight characters are taken from the 1st eight non-blank and non-punctuation characters
b) to create a multi-line message description instead of SSWENCODE 1.0;ASCII.

Syntax: SSW_Header_Compatibility {No|Yes}

Default: SSW_Header_Compatibility No

Example: SSW_Header_Compatibility Y

SSWDA Program customization

The SSWDA program dynamically allocates the SYSOUT dataset used by OP/SS for its output as SYSOUT=A. You can change the JES SYSOUT class used. You can also customize SSWDA to force

- a) all messages with DOCUMENTTYPE=FINL to be sent as attachments or
- b) all messages with DOCUMENTCLASS=FINALDOC to be sent as attachments

These customizations are performed by updating the SSWOPTS member in INSTLIB and then running the INSTLIB job OPSISSWD.

Customizing user exits

Performing your own distribution list processing

SMTP-Send allows you to use distribution lists either through the Configuration file or using [LDAP](#). If you need to perform your own expansion of distribution lists, you can also do so using the distribution list expansion user exit.

You specify the expansion exit program name via the DISTLISTPGM Configuration statement. As each e-mail address is encountered in the input control statements it is inspected for the @ character that separates the host from the domain. If no @ is found, the e-mail address is eligible for distribution list processing and is passed to the expansion exit program for expansion. Each address returned by the distribution list exit program will still be passed to the address validation exit for validation.

The expansion exit program name on the DISTLISTPGM statement is restricted to 7 characters. When SMTP-Send runs in MVS Batch or TSO the character 'B' is appended to the program name. Under CICS, the character 'C' is appended. For example, under CICS the statement DISTLISTPGM DISTLUX causes program DISTLUXC to be used to expand distribution lists.

You can pass parameters to your expansion exit by adding them to the DISTLISTPGM Configuration statement following the program name. Enclose the entire value in double quotes as follows:

```
DISTLISTPGM "DISTLUX,IPAddress=127.0.0.0"
```

The address of the string `IPAddress=127.0.0.0` will be passed to the expansion exit in the passed data structure along with the length of the string. If the length is zero, no parameters were supplied.

The parameters passed to the DISTLISTPGM program are stored in plain text in the Configuration file. If you need to pass a password value to your exit program, you can code an asterisk character, *, in place of the password in your parameter string. SMTP-Send will use the value from the `Exit_Password` or `LDAP_Password` configuration statements as appropriate in place of the * when your exit is invoked.

Sample S/390 Assembler expansion exit programs for Batch and CICS are in INSTLIB members DISTLUXB and DISTLUXC.

Parameter List

The exit program is passed the following data structure:

Field	Description
address[4]	Contains a pointer to a 12-byte workarea that is returned after each call to the user exit. This can be used to retain information between calls.
int[4]	Contains the return code from the user exit. This is in addition to the value returned via R15, since CICS applications are unable to return values in registers.
address[4]	Contains a pointer to any parameters that may have been supplied on the DISTLISTPGM control statement. If no parameters were provided, the pointer is zero.
byte[1]	Contains the length of the parameters supplied on the DISTLISTPGM control statement. If no parameters were provided, the length is zero.
byte[1]	Contains the length of a message returned by the user exit. If no message is returned this value should be set to zero. The maximum length that can be displayed by SMTP-Send is 125 characters.
int[2]	Contains a binary number with the sequence number of the member within the distribution list.
character[125]	Contains a message from the user exit to be displayed by SMTP-Send.
byte[1]	On entry to the user exit contains the length of the SMTP name passed to the user exit. On return it contains the length of the SMTP name returned by the user exit.
character[128]	Contains the distribution list name. The user exit should use this field to return a member of the distribution list.
byte[1]	Reserved.
byte[1]	x'00' or x'40' indicates V1 format, C'1' indicates V2 format and presence of DLDATA extension
int[2]	time in seconds to wait for a response from a LDAP search request
int[2]	number of times to attempt to connect to the LDAP server
int[2]	time in seconds to wait between LDAP connection attempts

Start of DLDATA Extension

address[4] pointer to encryption routine supplied by SMTP-Send. This routine can be used by your exit to encrypt a password value before calling LDAP.

address[4] pointer to decryption routine. Supplied by SMTP-Send on calls to LDAP or by user exits when calling LDAP from within an exit. Zero indicates that the password below is in plain text.

When calling encryption or decryption programs, pass the address of the following structure in General Register 1

byte[1] length of plain text password before encryption or after decryption.

byte[1] length of encrypted password after encryption or before decryption.

Field	Description
character[128]	plain text password value. Place value to be encrypted here before calling encryption program. Value is replaced by decryption program
byte	reserved
character[128]	encrypted text password value. Place value to be decrypted here before calling decryption program. Value is replaced by encryption program. If the name of the expansion exit program name on the DISTLISTPGM statement begins with LDAP, the LDAP_Password is copied into this field; otherwise, the Exit_Password value is copied here.
byte	reserved

You need to check if the distribution list name passed to your exit is that of an existing distribution list. If it is then the user corresponding to the sequence number, i.e. 1, 2, etc, should be returned. The exit is called repeatedly until a non-zero return code is returned.

Under MVS/Batch or TSO, the data structure is passed by address. SMTP-Send will LINK to the exit program with Register 1 pointing to the address of an address that points to a 2-Byte length followed by a 9-Character string representation of the data structure address in decimal.

For example, if the data structure is located at X'201616A8' (538318504 in decimal). SMTP-Send will build the following string: X'0009F5F3F8F3F1F8F5F0F4' and store the address of that string in its working storage and then load the value of that working storage location into Register 1.

Under CICS, the data structure is passed as a CICS Commarea.

Return Codes

The called program should return:

Return Code	Meaning
0	a name is returned for the given distribution list and the user exit should be called again
4	name passed is not a distribution list or no more members exist in the list

Performing your own address validation

SMTP-Send allows you to perform address validation in an exit program. You can use this exit to prevent unauthorized users from pretending they are someone else.

You specify the exit program name via the VALIDATEPGM Configuration statement. When a program name has been specified, every prospective SMTP address is passed to your exit program along with additional descriptive information before it is used. You can accept the address passed by SMTP-Send, replace the SMTP name with a new value or reject the SMTP name.

The validation exit program name on the VALIDATEPGM statement is restricted to 7 characters. When SMTP-Send runs in MVS Batch or TSO the character 'B' is appended to the program name. Under CICS, the character 'C' is appended. For example, under CICS a VALIDATEPGM VALIDP statement causes the program VALIDPC to be used to provide address validation logic.

You can pass parameters to your validation exit by adding them to the VALIDATEPGM Configuration statement following the program name. Enclose the entire value in double quotes as follows:

```
VALIDATEPGM "VALIDP,IPAddress=127.0.0.0"
```

The address of the string `IPAddress=127.0.0.0` will be passed to the expansion exit in the passed data structure along with the length of the string. If the length is zero, no parameters were supplied.

The parameters passed to the VALIDATEPGM program are stored in plain text in the Configuration file. If you need to pass a password value to your exit program, you can code an asterisk character, *, in place of the password in your parameter string. SMTP-Send will use the value from the `Exit_Password` configuration statement in place of the * when your exit is invoked.

Sample S/390 Assembler validation exit programs for Batch and CICS are in INSTLIB members VALIDPB and VALIDPC.

If required you can call the TBS LDAP routines from your validation exit to extract any additional data you might need to validate the address. Refer to [Using LDAP](#) for more information.

A note on APF authorization

If you plan to use security services that require APF authorization under MVS Batch or TSO, then you must (a) relink the SMTP-Send load module with the AC(1) attribute and (b) authorize the SMTP-Send load library. See [APF authorization](#) in the Installing section.

Parameter List

The exit program is passed the following data structure (mapped by INSTLIB member VALIDATA) either via standard OS linkage (pointed to by R1) or as a CICS commarea:

Field	Description
int[4]	Contains the return code from the user exit. This is in addition to the value returned via R15, since CICS applications are unable to return values in registers.
address[4]	Contains a pointer to a 12-byte workarea that is returned after each call to the user exit. This can be used to retain information between calls.
address[4]	Contains a pointer to any parameters that may have been supplied on the VALIDATEPGM control statement. If no parameters were provided, the pointer is zero.
byte[1]	Contains the length of any parameters that may have been supplied on the VALIDATEPGM control statement. If no parameters were provided, the length is zero.
character[8]	The current jobname.
character[8]	RACF or CICS userid of the current user.
character[80]	Default domain specified during SMTP-Send customization. The user exit can use this value to create addresses that use the default domain when no domain can be determined from the input data.
byte[1]	Contains the length of a message returned by the user exit. If no message is returned this value should be set to zero. The maximum length that can be displayed by SMTP-Send is 125 characters.
character[125]	Contains a message returned from the user exit to be displayed by SMTP-Send.
character[1]	Type of SMTP address: 'T' - TO address 'S' - sender's address 'C' - CC address 'B' - BCC address 'R' - REPLY_TO address
character[4]	Reserved. OP/SS pointer to DGN (address) value.
character[4]	Reserved. OP/SS length of DGN.
character[4]	Reserved. OP/SS pointer to DEN (user) value.
character[4]	Reserved. OP/SS length of DEN.
character[128]	Original recipient data as entered on control card; the user exit can use this data to validate the address.
byte[1]	On entry to the user exit, contains the length of the SMTP name passed to the user exit. On return, it contains the length of the SMTP name returned by the user exit.
character[128]	Contains the proposed SMTP name. The user exit can use this field to return an alternate SMTP address.
byte[1]	Filler byte used to terminate the proposed SMTP name for C and C++ callers.
byte[1]	Indicates that the userid of the sender has been obtained from RACF/SAF during address lookup due to a configuration option.
int[4]	Contains the length of the Original recipient data.
address[4]	Contains a pointer to the LDAP exit program parameter string passed on the LDAP_LOOKUP_PGM statement. Contains zero if no parm or LDAP exit was specified.
byte[1]	Contains the length of the LDAP exit parameter or zero if no LDAP exit parm was specified.
byte[1]	Flags to indicate the results of address lookups for this address: x'01' - no lookup performed x'02' - LDAP lookup failed x'04' - address mapping exit lookup failed x'08' - default domain has been appended to address
byte[1]	Controls subsequent invocation of data manipulation exit. Any non-zero value indicates the exit will be called. X'FF' indicates that the ALWAYSCALLDATAEXIT statement in the SMTP-Send

Field	Description
	Configuration dataset requests that the exit always be called. You can clear this field if you do not want the exit to be called.
address[4]	Contains a pointer to a 12-byte workarea that is passed between the validation and data manipulation exits. This can be used to retain information between calls.
character[6]	Contains the 6 character prefix of the LDAP lookup program from the LDAP_LOOKUP_PGM OPSCONFG statement padded on the right with nulls.
byte[1]	reserved

Under MVS/Batch or TSO, the data structure is passed by address. SMTP-Send will LINK to the exit program with Register 1 pointing to the address of an address that points to a 2-Byte length followed by a 9-Character string representation of the data structure address in decimal.

For example, if the data structure is located at X'201616A8' (538318504 in decimal), SMTP-Send will build the following string: X'0009F5F3F8F3F1F8F5F0F4' and store the address of that string in its working storage and then load the value of that working storage location into Register 1.

Under CICS, the data structure is passed as a CICS Commarea.

Return Codes

The called program should return:

Return Code	Meaning
0	if the proposed name is acceptable
4	the original SMTP value was replaced by a new value in the SMTP name field
8	if the proposed name is declined by the exit and must not be used
16	if the new name will not fit in the 128 character destination field

Reformatting data before it is sent via SMTP

SMTP-Send allows you to manipulate the data assembled by SMTP-Send before it is sent. You can use this exit to perform any additional data substitution you might require.

You specify the exit program name via the DATAEXITPGM Configuration statement. When a program name has been specified, the exit is called immediately before the data is sent. You cannot change the mail headers from this exit; the From, To, Cc, Bcc and Subject information is read-only.

The exit program name on the DATAEXITPGM statement is restricted to 7 characters. When SMTP-Send runs in MVS Batch or TSO the character 'B' is appended to the program name. Under CICS, the character 'C' is appended. For example, under CICS, the statement, DATAEXITPGM DATAXT causes the program DATAXTPC to be used to provide data manipulation logic.

You can pass parameters to your exit by adding them to the DATAEXITPGM Configuration statement following the program name. Enclose the entire value in double quotes as follows:

```
DATAEXITPGM "DATAXT, IPAddress=127.0.0.0"
```

The address of the string IPAddress=127.0.0.0 will be passed to the expansion exit in the passed data structure along with the length of the string. If the length is zero, it means that no parameters were supplied. The data is contained in a dataset for which the DDname is passed in the exit data structure. You must open the dataset and read the data in your exit. You can replace the data by allocating a new dataset and returning the new DDname to be used.

Sample S/390 Assembler data manipulation exit programs for Batch and CICS are in INSTLIB members DATAXTB and DATAXTC. Refer to them for details on accessing the header information.

A note on APF authorization

If you plan to use services that require APF authorization under MVS Batch or TSO, then you must (a) relink the SMTP-Send load module with the AC(1) attribute and (b) authorize the SMTP-Send load library. See [APF authorization](#) in the Installing section.

Parameter List

The exit program is passed the following data structure (mapped by INSTLIB member DATACOMM) either via standard OS linkage (pointed to by R1) or as a CICS commarea:

Field	Description
int[4]	Contains the return code from the user exit. This is in addition to the value returned via R15, since CICS

Field	Description
	applications are unable to return values in registers.
address[4]	Contains a pointer to a 12-byte workarea that is returned after each call to the user exit. This can be used to retain information between calls.
address[4]	Contains a pointer to any parameters that may have been supplied on the DATAEXITPGM control statement. If no parameters were provided, the pointer is zero.
byte[1]	Contains the length of any parameters that may have been supplied on the DATAEXITPGM control statement. If no parameters were provided, the length is zero.
character[8]	The current jobname.
character[8]	RACF or CICS userid of the current user.
address[4]	Contains a pointer to the linked list of From information after all address lookups and validation have completed.
address[4]	Contains a pointer to the linked list of To information after all address lookups and validation have completed. Zero if no To information has been provided.
address[4]	Contains a pointer to the linked list of Cc information after all address lookups and validation have completed. Zero if no Cc information has been provided.
address[4]	Contains a pointer to the linked list of Bcc information after all address lookups and validation have completed. Zero if no Bcc information has been provided.
address[4]	Contains a pointer to the linked list of Subject information after all address lookups and validation have completed. Zero if no Subject information has been provided.
byte[1]	On entry to the user exit, contains the length of the SMTP name passed to the user exit. On return, it contains the length of the SMTP name returned by the user exit.
character[8]	Contains the DDname for the dataset or 8-byte CICS Temporary Storage Queue name containing the data to be sent.
character[1]	Indicates whether the data is in a MVS file 'F' or CICS TS Queue 'T'.
byte[1]	Contains the length of a message returned by the user exit. If no message is returned this value should be set to zero. The maximum length that can be displayed by SMTP-Send is 125 characters.
character[125]	Contains a message returned from the user exit to be displayed by SMTP-Send.
address[4]	Contains a pointer to the LDAP exit program parameter string passed on the LDAP_LOOKUP_PGM statement. Contains zero if no parm or LDAP exit was specified.
byte[1]	Contains the length of the LDAP exit parameter or zero if no LDAP exit parm was specified.
byte[1]	Indicates how data exit was invoked. X'FF' indicates data invoked due to ALWAYSALLDATAEXIT in the SMTP-Send Configuration dataset. Any other value is set by the address validation exit.
address[4]	Contains a pointer to a 12-byte workarea that is passed between the validation and data manipulation exits. This can be used to retain information between calls.
character[6]	Contains the 6 character prefix of the LDAP lookup program from the LDAP_LOOKUP_PGM OPSCONFG statement padded on the right with nulls.
byte[1]	reserved

Under MVS/Batch or TSO, the data structure is passed by address. SMTP-Send will LINK to the exit program with Register 1 pointing to the address of an address that points to a 2-Byte length followed by a 9-Character string representation of the data structure address in decimal.

For example, if the data structure is located at X'201616A8' (538318504 in decimal). SMTP-Send will build the following string: X'0009F5F3F8F3F1F8F5F0F4' and store the address of that string in its working storage and then load the value of that working storage location into Register 1.

Under CICS, the data structure is passed as a CICS Commarea.

Return Codes

The called program should return:

Return Code	Meaning
0	the data should be sent as is
8	the distribution should be aborted and the data should not be sent

OP/SS Performing your own address mapping

OP/SS allows you to perform your own address mapping in an exit program. You can replace the address mapping logic supplied by OP/SS or supplement it by adding your own mapping logic, for example to interrogate an external directory.

You specify the address mapping exit program name via the MAPPINGPGM Configuration statement. When a program name is specified, every SMTP address is passed to your exit program along with additional descriptive information before it is used. You can lookup the SMTP name based upon the input value or accept the address suggested by OP/SS.

The mapping exit program name on the MAPPINGPGM statement is restricted to 7 characters. When OP/SS runs in MVS Batch or TSO the character 'B' is appended to the program name before it is used. Under CICS, the character 'C' is appended. For example, under CICS the statement MAPPINGPGM LOOKUP causes program LOOKUPC to be used to provide custom address mapping logic.

You can pass parameters to your address mapping exit by adding them to the MAPPINGPGM Configuration statement following the program name. Enclose the entire value in double quotes as follows:

```
MAPPINGPGM "LOOKUP,IPAddress=127.0.0.0"
```

The address of the string `IPAddress=127.0.0.0` is passed to the address mapping exit in the passed data structure along with the length of the string. A length of zero indicates that no parameters were supplied.

You can pass an encrypted password value to your exit by specifying an `Exit_Password` OPSCONFG control statement. The address of a decryption routine will be passed along with the password in the MAPPRDAT data structure.

Sample mapping validation programs written in S/390 Assembler can be found in INSTLIB members LOOKUPB and LOOKUPC. You can also call the TBS LDAP routines from your validation exit to extract any additional data you might need to lookup the address. Refer to [Using LDAP](#) for more information.

The exit program is passed the following data structure either via standard OS linkage (pointed to by R1) or as a CICS commarea:

Field	Description
int[4]	Contains the return code from the user exit. This is in addition to the value returned via R15, since CICS applications are unable to return values in registers.
address[4]	Contains a pointer to a 12-byte workarea that is returned after each call to the user exit. This can be used to retain information between calls.
address[4]	Contains a pointer to any parameters that may have been supplied on the MAPPINGPGM control statement. If no parameters were provided, the pointer is zero.
byte[1]	Contains the length of any parameters that may have been supplied on the MAPPINGPGM control statement. If no parameters were provided, the length is zero.
character[8]	The current jobname.
character[8]	RACF or CICS userid of the current user.
character[80]	Default domain specified during OP/SS customization. The user exit can use this value to create addresses that use the default domain when no domain can be determined from the input data.
byte[1]	Contains the length of a message returned by the user exit. If no message is returned this value should be set to zero. The maximum length that can be displayed by OP/SS is 125 characters.
character[125]	Contains a message returned from the user exit to be displayed by OP/SS.
character[1]	Intent for DGN.DEN value in parms 2 and 3, 'T' - TO address, 'S' - sender's address, 'C' - CC address, 'B' - BCC address, 'R' - REPLY_TO address
character[4]	pointer to DGN (address) value.
character[4]	length of DGN.
character[4]	pointer to DEN (user) value.
character[4]	length of DEN.
character[128]	Original recipient data as entered on control card. The user exit can use this data to assist in the address mapping.
byte[1]	On entry to the user exit, contains the length of the SMTP name passed to the user exit. On return, contains the length of the SMTP name returned by the user exit.
character[128]	Contains the proposed SMTP name. The user exit can use this field to return an alternate SMTP address.
byte[1]	Filler byte used to terminate the proposed SMTP name for C and C++ callers.
byte[1]	x'00' or x'40' indicates V1 format, C'1' indicates V2 and presence of extension
int[4]	Contains the length of the original recipient data.
address[4]	Contains a pointer to the LDAP exit program parameter string passed on the LDAP_LOOKUP_PGM statement.
Byte[1]	Contains zero if no parm or LDAP exit was specified
Start of V2 Extension	Length of LDAP parameter string, or zero if none was specified.
address[4]	pointer to encryption routine supplied by SMTP-Send. This routine can be used by your exit to encrypt a password value before calling LDAP.
address[4]	pointer to decryption routine. Supplied by SMTP-Send on calls to LDAP or by user exits when calling LDAP from within an exit. Zero indicates that the password below is in plain text.
When calling encryption or decryption programs, pass the address of the following structure in General Register 1	
byte[1]	length of plain text password before encryption or after decryption.
byte[1]	length of encrypted password after encryption or before decryption.
character[128]	plain text password value. Place value to be encrypted here before calling encryption program. Value is replaced by decryption program
byte	reserved
character[128]	encrypted text password value. Place value to be decrypted here before calling decryption program. Value is replaced by encryption program.
	The <code>Exit_Password</code> value is copied here.
byte	reserved
byte	reserved

Field	Description
character[6]	Contains the 6 character prefix of the LDAP lookup program from the LDAP_LOOKUP_PGM OPSCONFG statement padded on the right with nulls.
byte	reserved

Under MVS/Batch or TSO, the data structure is passed by address. SMTP-Send will LINK to the exit program with Register 1 pointing to the address of an address that points to a 2-Byte length followed by a 9-Character string representation of the data structure address in decimal.

For example, if the data structure is located at X'201616A8' (538318504 in decimal). SMTP-Send will build the following string: X'0009F5F3F8F3F1F8F5F0F4' and store the address of that string in its working storage and then load the value of that working storage location into Register 1.

Under CICS, the data structure is passed as a CICS Commarea.

Return Codes

The called program should return:

Code	Meaning
0	if the proposed name is acceptable
4	the original SMTP value was replaced by a new value in the SMTP name field
8	if an SMTP name could not be determined from the DGN and DEN values
16	if the new name will not fit in the 128 character destination field

Changing SMTP-Send messages

Translating messages into a different language

SMTP-Send messages are stored in a message table in INSTLIB. The message table is named OPSSTR0. To translate the SMTP-Send messages into your own language run INSTLIB job OPSCUSTR. This job creates the appropriate load module.

How SMTP-Send messages are structured

SMTP-Send messages consist of:

Fixed text parts Fixed text parts stay the same each time SMTP-Send generates the message. They consist of letters and numbers that you can change using the message definition macro #TBSSTRD. This macro is described below.

Variable text parts Variable text parts may differ each time SMTP-Send generates the message. These parts depend on the information that SMTP-Send is processing when the message is generated.

Insertion points Insertion points indicate where variable text parts will be inserted in the message text.

The message definition macro #TBSSTRD

Each text message that SMTP-Send generates is defined using the #TBSSTRD macro. To change a text message, edit the #TBSSTRD macro statement for that message. Follow the syntax shown below to code the messages you want to change.

```
[label] #TBSSTRD num, 'text', MAXLEN=(m1, l1, . . . , l4)
```

label (optional) A unique label for the #TBSSTRD macro. Specify one to eight alphanumeric characters, beginning in column one. The first character must be a letter.

num The message number. Do not change this value.

text The message text, which consists of fixed text parts and up to four insertion points. You can change this text.

(*m1*, *l1*, . . . , *l4*) The maximum length for the message.. If the changed message will exceed this length, shorten the message. Do not change the value of MAXLEN.

The value of *m1* is the total maximum length. If the message contains insertion points (described below), these insertion points are represented by *l1* to *l4*. A 79-character message with two insertion points, one of 5 characters and one of 3 characters, is specified as:
MAXLEN=(79, 5, 3)

If there are no insertion points, you do not need the brackets or the values *l1* to *l4*. A 79-character message with no insertion points is specified as:
MAXLEN=79

Coding message text

Quotation marks

Message text consists of a character message enclosed in single quotation marks ('). Because the single quotation mark is used to enclose text, you must use two single quotation marks (") where you want a single quotation mark to appear within the text.

Insertion points

Message text may also include insertion points specified as &?. These insertion points tell SMTP-Send where to insert variable information. The value of # ranges from 1 to 4. You can change the text around an insertion point, or the location of an insertion point, but do not change the value of #.

The number of insertion points must be less than or equal to the number of insertion points in the original text. If you use fewer insertion points than were in the original text, the message will lose the information conveyed by the omitted insertion point(s).

Long messages

Do not put message text in columns 72 to 80, because this text is ignored. If a message will not fit in a single line, put any character in column 72, then begin the next line in column 16.

Special characters in languages other than English

SMTP-Send assumes that all text in message tables is in code page 500. If you are translating messages to a language other than English, make sure that you use code page 500 for special characters such as #, \ and Ó

Customizing ISPF panel support

ISPF dialogs have been included to enable TSO/ISPF users to invoke SMTP-Send from ISPF. The SMTP-Send dialogs are unloaded using job OPSI09.

Installing the SMTP-Send dialogs using standard ISPF libraries

To install the SMTP-Send ISPF dialogs either concatenate the SMTP-Send ISPF libraries with the standard ISPF libraries OR copy the SMTP-Send ISPF library members into the corresponding standard ISPF libraries. For example, copy the members in prefix.ISPPLIB into the standard ISPPLIB library. With the exception of the main SMTP-Send CLIST, SMTPSEND, all SMTP-Send ISPF library members begin with OPS, so there will be no overlap with any existing ISPF members.

The ISPF installation includes ISPF panels, messages and CLIST libraries. These files must be accessible to users before they can run the SMTP-Send dialogs.

If you choose not to copy the SMTP-Send ISPCLIB members into your standard ISPPLIB library or a library already in your SYSPROC concatenation, you can instruct SMTP-Send to use a Application ALTLIB to inform ISPF where to locate the OPS CLISTS. Alter the CLISTS OPSMENU and SMTPSEND, specifying your CLIST dsname in the CLIB parameter on the first line of each CLIST. The dsname will be surrounded by quotes before use, so none are necessary in the parameter value.

Once the members have been copied into the appropriate libraries, you can invoke SMTP-Send using the command TSO OPSMENU. To add the dialogs to an ISPF panel do the following:

Step	Procedure
1	Modify the panel to add a menu item for SMTP-Send. For example: <pre>15 - Send mail to the Internet</pre>
2	Add the SMTP-Send panel to the selection list. For example: <pre>&ZSEL = TRANS (TRUNC (&ZCMD, '.') . 15, 'CMD(OPSMENU) NEWAPPL(OPS) ' . .</pre>

Customizing the SMTP-Send dialogs

The CLIST OPSDEFLT sets the various default values used by SMTP-Send. Although dialogs are provided for users to alter these values themselves, it is recommended that you update this CLIST to set the correct values at installation time. Values you need to specify include the data set names of your SMTP-Send Configuration file and LOADLIB and the domain name for your enterprise.

Using the TBS SMTP Spool Server

The SMTP Spool Server, OPSSSRVR, allows SMTP data written to the JES spool to be relayed to a SMTP relay server. This spool data can be written by SMTP-Send as part of recovery from TCP/IP failures or written by other applications such as SMTPNOTE or XMITIP destined to be processed by IBM's SMTP Server.

Overview

The SMTP Spool Server is a functional replacement for the outbound capabilities of IBM's SMTP server started task.

The administrator controls input selection of SYSOUT datasets using the external writer name, and optionally the class, and destination parameters in the Configuration VSAM file. Other configuration parameters are also available to control OPSSSRVR.

Note: This can be a separate file from the one used to configure SMTP-Send or the entries can be added to your existing file using the OPSSSRVR configuration program.

Non-held JES output that matches the input selection criteria is selected for processing.

If communications fail with the receiving SMTP relay server(s), OPSSSRVR waits on a timer for a specified interval in the configuration file. Failure results in the distributions being left on the JES SPOOL until such time as an SMTP server becomes available or the retry count is exceeded. In the event the retry count is exceeded, an operator message is written to the console requesting a reply to either repeat the retry logic or to terminate OPSSSRVR. If no SMTP servers will be available for some time, the operator should terminate OPSSSRVR. Once the receiving SMTP server is available, the operator should restart OPSSSRVR.

If the receiving SMTP server rejects the incoming message, the SMTP data stream is sent as an attachment in a non-delivery message to the configured recipient.

When there is no work to process, OPSSSRVR goes into a wait state until one of three events occurs:

1. A SYSOUT dataset becomes available that matches the input selection criteria.
2. A system command is received from the console operator.
3. Retry mode is in effect due to communications failure with the SMTP server(s) and the delay interval timer expires, meaning it's time to try again.

OPSSSRVR supports single or multiple distributions in a single SYSOUT dataset

OPSSSRVR supports JES Network format data such as that produced by IBM's SMTPNOTE REXX commands. Use the configuration parameters to select SYSOUT datasets produced by SMTPNOTE.

Send details may be logged to the MVS SMF datasets or to an optional user supplied sequential dataset. Details logged include the original job name and Userid, the document size, date, sender, reply-to, subject, all recipients, and any attachment names if applicable.

Note: you can use a different SMF ID to combine your OPSSSRVR log records with those from SMTP-Send.

A user customizable message table, OPSMSGTx is used, similar to that used by SMTP-Send.

LOG messages are written to a common file named OPSSLOG and other messages such as debugging messages are written to OUTPUT. The Modify operator command can be used to change a number of application control parameters. The START operator command can be used to initiate OPSSSRVR provided the proper JCL has been added to PROCLIB, and the STOP operator command can be used to terminate OPSSSRVR.

Configuration Parameters

Using the SMTP Spool Server

The following configuration parameters are used to control various aspects of OPSSSRVR as indicated. Where a control statement is common between SMTP-Send and the Spool Server the complete definition is not repeated in this section.

ATSIGN

Same as SMTP-Send [ATSIGN](#) Configuration statement

BATCH_AUTHCODE

Same as SMTP-Send [BATCH_AUTHCODE](#) Configuration statement

DEBUG

Purpose: Messages with debugging information are written to the OPSSLOG and OUTPUT datasets if this value is coded. YES provides some detail, VERBOSE significantly more, while TRACE provides the most amount of extra detail regarding current processing.

Syntax: DEBUG {YES|NO|VERBOSE|TRACE}

Default: DEBUG NO

Example: DEBUG YES

Note: to enable DEBUG only for a particular job, add PARM='value' on the EXEC PGM=OPSGS3B JCL statement

DEFAULT_DOMAIN

Same as SMTP-Send [DEFAULT_DOMAIN](#) Configuration statement

ERROR_MSG_FROM_NAME

Same as SMTP-Send [ERROR_MSG_FROM_NAME](#) Configuration statement

ERROR_MSG_TO_NAME

Same as SMTP-Send [ERROR_MSG_TO_NAME](#) Configuration statement

FAIL_ON_ERROR

Same as SMTP-Send [FAIL_ON_ERROR](#) Configuration statement

FIXSPOOLLEADINGDOTS

Purpose: Instructs OPSSSRVR to apply the SMTP leading dot transparency feature to data. When a receiving SMTP server encounters a line with a length greater than 1 that begins with a `.` character it will remove the character, decreasing the length of the incoming line by 1. When FixSpoolLeadingDots is set to Yes, OPSSSRVR will insert an extra `.` character at the beginning of each line longer than 1 character that begins with a `.` More information on the SMTP leading dot transparency feature can be found in the SMTP specification, RFC 2821 Section 4.5.2

NOTE: This specification should only be used when SMTP streams are created outside of the SMTP-Send environment and may contain `.` characters at the beginning of lines, such as used by some users for document formatting. A specification of All will cause an extra `.` to be inserted even for lines containing only a single `.` character. This should only be used where the SMTP input streams contain single distributions as single `.` intended to terminate a distribution will instead be considered to be data

Syntax: FixSpoolLeadingDots {YES|NO|ALL}

Default: FixSpoolLeadingDots NO

LOGGING

Same as SMTP-Send [LOGGING](#) Configuration statement

RELAY_RETRY_COUNT

Purpose: Number of times OPSSSRVR will attempt to resend a distribution before requesting operator intervention

Syntax: RELAY_RETRY_COUNT [*nnn*|3]

Default: RELAY_RETRY_COUNT 3

Example: RELAY_RETRY_COUNT 5

RELAY_RETRY_DELAY

Purpose: Time in minutes to wait before attempting to resend a distribution or look on spool for another message to be relayed

Syntax: RELAY_RETRY_DELAY [*nnn*|10]

Default: RELAY_RETRY_DELAY 10

Example: RELAY_RETRY_DELAY 5

RELAY_LOG_DETAILS

Purpose: tells OPSSSRVR to display the sender and recipient names of relayed e-mails in the OPSSLOG data set. The name of the data set from the JES Spool is also displayed.

Syntax: Relay_Log_Details [YES|NO]

Default: Relay_Log_Details NO

Example: Relay_Log_Details YES

RELAY_SMF_TYPE

Purpose: specifies the SMF record type (or number) that contains log data. In batch, this is a value between 129 and 255.

Syntax: Relay_SMF_TYPE [*nnn* | 129]

Default: Relay_SMF_TYPE 129

Example: Relay_SMF_TYPE 255

REMOVETRAILINGBLANKS

Same as SMTP-Send [REMOVETRAILINGBLANKS](#) Configuration statement

SMTP_RELAY_ADDRESS

Same as SMTP-Send [SMTP_RELAY_ADDRESS](#) Configuration statement

SMTP_RELAY_ADDRESS_ALTERNATE

Same as SMTP-Send [SMTP_RELAY_ADDRESS_ALTERNATE](#) Configuration statement

SMTP_SERVER_CLASS

Same as SMTP-Send [SMTP_SERVER_CLASS](#) Configuration statement

SMTP_SERVER_NAME

Same as SMTP-Send [SMTP_SERVER_NAME](#) Configuration statement

SMTP_SERVER_NODE

Same as SMTP-Send [SMTP_SERVER_NODE](#) Configuration statement

USECUSTOMCODEPAGE

Same as SMTP-Send [USECUSTOMCODEPAGE](#) Configuration statement

WTO_onTCPFailure

Same as SMTP-Send [WTO_onTCPFailure](#) Configuration statement

Installation Considerations

The Server mainline program, OPSSSRVR, must be linked SETCODE AC(1) and reside in an APF authorized library in order to read the JES SPOOL using the Subsystem Interface. Note that all other libraries used in STEPLIB must also be APF authorized.

Example: `SETPROG APF,ADD,DSN=OPS.LOADLIB,VOLUME=vvvvvv`

If not APF-authorized, you will receive the message, OPS104E JES Subsystem Interface error. RC=0072, RSN=0000, and SMTP- Send will terminate with a return code of 8,

Logging

Processing messages are written to a sequential file named OPSSLOG.

Each message is preceded by the Julian date YYDDD followed by the time HH:MM:SS

The message number ends with a type (column 23) which indicates one of the following:

- I** Informational message
- W** Warning message
- E** Error message
- D** debug message

Sample OPSSLOG:

```
02227 14:45:04 OPSSSRVR (c) COPYRIGHT 2002,2005 BY TBS SOFTWARE INC., ALL RIGHTS RESERVED
02227 14:45:04 OPS001I OPSSSRVR Initialization - Version R5.1 PTF 1A
02227 14:45:04 OPS006I External writer name is SMTP
02227 14:45:04 OPS016I External writer class is X
02227 14:45:04 OPS007I Retry delay interval is 005 minutes
02227 14:45:04 OPS008I Retry count is 02
02227 14:45:04 OPS009I SMTP relay address is 127.0.0.1:8026
02227 14:45:04 OPS010I SMTP alternate relay address is 127.0.0.1:25
02227 14:45:04 OPS024I NDR's sent to: POSTMASTER@TBSSOFT.COM
02227 14:45:04 OPS025I NDR's sent from: SMTPSEND@TBSSOFT.COM
02227 14:45:04 OPS027I Default Domain: TBSSOFT.COM
02227 14:45:04 OPS019I AtSign Character is: @
02227 14:45:04 OPS012I SMF logging to record type 130 is active
02227 14:45:04 OPS028I GetHostByName is: 0
02227 14:45:04 OPS029I FailOnError is: 0
02227 14:45:04 OPS036I RemoveTrailingBlanks is 1
02227 14:45:04 OPS032I Relay_Log_Details is active
02227 14:45:04 OPS001I OPSSSRVR 14.42 11/30/05 Initialized
02227 15:45:43 OPS004I Processing dataset TSSJES1.JOB04887.D0000103.0
02227 15:45:44 OPS020I Received Note from <demo@tbssoft.com>
02227 15:45:44 OPS021I Delivering Note to <support@tbssoft.com>
02227 15:45:44 OPS021I Delivering Note to <demo@tbssoft.com>
```

```
02227 15:45:45 OPS005I Dataset sent successfully
02227 15:48:38 OPS003I Received STOP operator command
02227 15:48:38 OPS002I OPSSSRVR is terminating
```

Running under MVS

The following sample JCL can be used to run OPSSSRVR

```
//OPSSSRVR JOB
//STEP1 EXEC PGM=OPSSSRVR
//STEPLIB DD DISP=SHR,DSN=OPS.LOADLIB
//OPSCONFG DD DSN=OPS.CONFIG,DISP=SHR
//OPSSLOG DD SYSOUT=*
//OUTPUT DD SYSOUT=*
```

EXEC runtime parameter

Override the DEBUG control statement in the CONFIG file using one of the NODEBUG, DEBUG, VERBOSE or TRACE values for the OPSSSRVR EXEC parameter.

Note: this parameter can also be controlled during execution using the Modify operator command as described below.

```
//STEP1 EXEC PGM=OPSSSRVR,PARM=DEBUG
```

Console Operator Commands

Initiate OPSSSRVR using the START Operator command:

```
example: S OPSSSRVR
```

Note: this assumes a valid member exists in SYS1.PROCLIB or other such library used for starting jobs.

Shutdown OPSSSRVR using the STOP Operator command:

```
example: P OPSSSRVR
```

Change execution parameters using the MODIFY command:

Terminate processing	F OPSSSRVR,QUIT
Turn ON basic debug messages	F OPSSSRVR,DEBUG
Turn OFF debug messages	F OPSSSRVR,NODEBUG
Turn ON verbose debug messages	F OPSSSRVR,VERBOSE
Turn ON debug trace-level messages	F OPSSSRVR,TRACE
Display configuration setting	F OPSSSRVR,SHOWCFG
Change the Spool Input class	F OPSSSRVR,CLASS= <i>c</i>
Toggle the SMTP leading dot transparency feature setting off or on (only on lines with >1 character)	F OPSSSRVR,FIXDOTS
Toggle the SMTP leading dot transparency feature setting off or on (for all data lines)	F OPSSSRVR,FIXALLDOTS
Change the amount of time OPSSSRVR waits between retries or requests to JES for additional messages	F OPSSSRVR,SLEEPINT= <i>nnn</i>
Change the alternate SMTP server to the specified IP name or address	F OPSSSRVR,SMTP= <i>server</i>
Change the Spool External Writer name	F OPSSSRVR,WTR= <i>nnnnnnnn</i>

Optional JCL statements

Add the following DD statement if you want to see the incoming SMTP data. You might find this option useful during testing.

```
//DATAIN DD SYSOUT=*
```

Messages and codes

Return Codes

Code	Meaning
0000	OPSSSRVR terminated without errors.
0004	Check the OPSSLOG for warning messages.
0008	Check the OPSSLOG for error messages.
0016	A severe error occurred and OPSSSRVR was terminated. Check OPSSLOG for details.

Code	Meaning
0032	The operator terminated OPSSSRVR when asked to reply, since there are no SMTP servers currently available.
0068	Error encountered loading TSSSEND module
0128	Internal logic error encountered

User Abends

U0001 Application is unable to write an operator message to the system console.

In the event OPSSSRVR is unable to communicate with the SMTP server(s), and the maximum retry count has been exceeded, an operator message is written to the console requesting what action to take.

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 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 *irecl*

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
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.
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

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.
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.

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.

Using LDAP

Overview

SMTP-Send allows you to expand distribution lists into e-mail addresses using LDAP (Lightweight Directory Access Protocol) compliant directories. LDAP is an Internet standard for accessing directory information using TCP/IP. Many popular e-mail systems such as Lotus Notes or Microsoft Exchange expose their directory information via LDAP.

SMTP-Send provides two types of LDAP lookup. You can choose to use either or both.

1. Expansion of distribution lists into one or more e-mail addresses
2. E-mail address lookup.

Expansion of distribution lists

Distribution lists contain a group of e-mail addresses and possibly other distribution lists. SMTP-Send uses LDAP to obtain all of the e-mail addresses in a distribution list by requesting each e-mail address in the group and then adding it to the appropriate addressing header. Duplicate e-mail addresses are removed from the expanded list.

E-mail address lookup

E-mail address lookup lets you obtain an e-mail address from your LDAP directory, for instance by looking up a Microsoft Exchange alias or Lotus Notes shortname. Unlike distribution list processing, e-mail address lookup is intended to return only a single address. If LDAP returns multiple entries, SMTP-Send considers this to be an error, displays an appropriate message, and skips the lookup for that e-mail address. You can also call SMTP-Send's LDAP address lookup routines from your own address validation, address lookup and data manipulation exit programs.

OP/SS With OP/SS, LDAP e-mail address lookups take one of two forms, userid and SNADS.

- Userid lookups are only performed for addresses specified using the Addressing extensions, FROM, TO, CC or BCC. Userid lookups will be performed for any e-mail address that does not contain a domain part (the part of the address following the @ character). The specified e-mail address will be used to perform a uid search as described below to obtain a complete e-mail address.
- SNADS address lookups are specified using a search filter. The search filter that OP/SS uses works with LDAP servers where the LDAP directory's uid field contains the SNADS user ID. If your LDAP directory stores SNADS user IDs in another field you can specify it using the search filter template that is described below. When you provide a search filter template, it is the only search filter that SMTP-Send uses.

Requirements and limitations

SMTP-Send requires OS/390 Release 5 or later in order to use LDAP.

SMTP-Send works with LDAP V2 and LDAP V3 servers.

Currently SMTP-Send supports only simple authentication. It does not support SSL binds to the LDAP server. If SMTP-Send connects to a V3 LDAP server, you can omit the bind name and password to perform an unauthenticated lookup.

LDAP client API

MVS Batch/TSO

In batch/TSO environments, SMTP-Send connects to LDAP servers using API code that is included with OS/390. This code, called the LDAP client API, is shipped as a set of load modules known as C/C++ DLLs. These DLLs are shipped in different formats depending on the release of OS/390. The LDAP routines of SMTP-Send are written using IBM C and require a Language Environment runtime of OS/390 R2.10 or later.

The LDAP client API is shipped in both PDS and HFS files. The PDS versions of the DLLs are normally installed into LPALIB as part of the OS/390 install. The section 'Compiling, Linking and Running a Program' in Chapter 1 of IBM book, LDAP Client Application Development Guide and Reference, (SC24-5878) details these requirements.

CICS

The SMTP-Send load library includes an LDAP API for CICS environments.

Testing your LDAP environment

IBM ships a MVS-specific version of the standard LDAP utility program, ldapsearch, you can use to test your LDAP client API implementation under TSO before you activate the LDAP support in SMTP-Send. The TSO command

```
ldapsrch -h IP_address_of_LDAP_Server -V 3 -s base -b "" "objectclass=*
```

provides information about the LDAP server named in the command. The result should look similar to this:

```
objectclass=top
subschemasubentry=cn=schema
namingcontexts=NOT Printable
supportedextension=1.3.6.1.4.1.1466.20037
supportedextension=LanguageCodes
supportedsaslmmechanisms=EXTERNAL
supportedldapversion=3
supportedldapversion=2
```

LDAPSRCH is described in the IBM book LDAP Server Administration and Usage Guide, (SC24-5861), section 'Running the LDAP Operation Utilities in TSO.' Use this book and the LDAP book referenced earlier in this chapter to diagnose any errors issued by this command.

Before you can run LDAPSRCH under TSO you must ensure that the following environment is set up:

1. PDS (GLDHLQ.SGLDLDP), where the LDAP Server load modules were installed, must be specified in one of LINKLIB, LPALIB or STEPLIB.
2. PDS (GLDHLQ.SGLDEXEC), which contains the CLISTs needed to run LDAPSRCH, must be available in SYSEXEC.

Activating LDAP lookups

LDAP processing is activated via SMTP-Send configuration statements. If migrating from a release prior to R5.2.0, refer to the Migrating section below.

To activate distribution list expansion, use the following SMTP-Send configuration statement:

```
LDAP_DISTLIST_PGM 'LDAPDL,IP address[:port|389] [,bind name][,{bind_password|"bind password" |*}][,D]'
```

To activate e-mail address lookup, use the following SMTP-Send configuration statement:

```
LDAP_LOOKUP_PGM 'LDAPAE,IP address[:port|389] [,bind name][,{bind_password|"bind password" |*}][,D],[template],[source_attribute]'
```

where:

Parameter	Description
IP Address	specifies the IP address of the LDAP server followed by the TCP/IP port number the LDAP server listens on. The IP address is a mandatory field and must be specified.
Bind name	specifies the bind distinguished name that is required to authenticate with the LDAP server. If no authentication is required you can skip this parameter and code only the comma. If the name contains spaces or commas, enclose it in double quotes.
Bind password	specifies the bind password for the bind name that is required to authenticate with the LDAP server. If no password is required you can skip this parameter and code only the comma. If the password contains spaces or commas, enclose it in double quotes. To avoid coding passwords in clear text in the configuration file, code an asterisk instead of the password. The LDAP exit will obtain the password from the LDAP_PASSWORD configuration statement which is stored in encrypted form.
DEBUG	The character 'D' causes the LDAP routines to produce diagnostic information. Data is written to the LDAPDEBUG DD, usually a SYSOUT dataset. Specifying 'V' instead of 'D' will produce verbose debugging information.
LDAP search filter template	SMTP-Send uses the filter template to perform lookups in custom fields that differ from the default UID template. Placeholders are used to describe the position where various runtime data will be inserted. %JOB - The jobname %JUID - The job or CICS transaction's User ID %DOMAIN - The default domain %DATA - The name from the addressing statement OP/SS OP/SS uses additional placeholders when a template is specified. These placeholders are replaced by values from the address lookup parameter list. The placeholders %DGN and %DEN indicate where the SNADS DGN and DEN (Address and Userid) values should be substituted. For example, a template of otherMailbox=SNADS\$%DGN(%DEN) could be used to lookup e-mail address for SNADS users in a Microsoft Exchange directory. The result is used as the LDAP search argument. If multiple fields are specified, enclose each one in parentheses, with no commas in between.

Parameter	Description
source LDAPAttribute	(optional, address lookup only) This specifies the LDAP attribute (aka directory field name), containing the data returned on address lookups. The default is 'mail'.

The total LDAP configuration information is limited to 256 characters in length by the OPSCONFG program.

The LDAP lookup is limited by the `LDAP_Timeout` SMTP-Send configuration statement. The LDAP server must respond within the specified number of seconds or the lookup will fail. The LDAP server may be too busy to accept requests when SMTP-Send makes its connection request. SMTP-Send will retry a finite number of times as requested via the `LDAP_OpenRetry` and `LDAP_OpenDelay` SMTP-Send configuration statements. `LDAP_OpenRetry` specifies the number of times to attempt to connect and `LDAP_OpenDelay` specifies the number of seconds to wait between attempts. If the connection to the LDAP server cannot be made, SMTP-Send ends with a return code of 16.

If the LDAP lookup for an email address fails after the LDAP server has been successfully contacted, SMTP-Send ends with a return code based on your configuration settings. Return code 6 means a message was sent because `FAIL_ON_ERROR` is NO. Return code 7xx means a message was not sent because `FAIL_ON_ERROR` is YES. The xx indicates which addressing component was in error.

CICS

To use LDAP functions under CICS you must define programs LDAPAE and LDAPDL. INSTLIB member OPSI03 contains the CICS definition statements that you will need.

Additional LDAP Configuration Statements

LDAP_OpenRetry {3 | count}

Purpose: Specifies the number of times the LDAP exit should attempt to connect to an LDAP server.

LDAP_OpenDelay {5 | seconds}

Purpose: Specifies the number of seconds to delay between attempts to connect to an LDAP server.

LDAP_Timeout {60 | seconds}

Purpose: Specifies the number of seconds to wait, before considering the LDAP request to have timed-out.

LDAP_Password value-

Purpose: Specifies the password value to be encrypted and stored in the OPSCONFG VSAM file. Specify a '*' character in the actual LDAP statements where the password is to be used.

Calling LDAP from your lookup, validation or data exit

SMTP-Send supports both name and directory lookups using LDAP. Only the name lookup function can be called from an exit. If you need to find data in LDAP that the base SMTP-Send code doesn't provide, you can call the SMTP-Send LDAP code to do the lookup for you. The simplest LDAP search is on the UID (UserID) attribute returning the mail (Internet e-mail) attribute value that matches the UID. You will probably request that SMTP-Send use a different LDAP attribute as the search target through a template. The LDAP lookup code will substitute supplied simple name values into the template, where substitution symbols are specified.

You can supply data for 3 substitution variables, %1, %2, %DATA to be substituted in your template. In addition, the current jobname, requesting user and default domain can be substituted via the %JOB, %JU and %DOMAIN symbols respectively as described in the LDAP Lookup parameter list.

Calling the LDAP lookup program

The first 6 characters of the LDAP lookup program name is passed to your exit via the LDAPPGMP variable in the commarea passed to your exit. The default value is LDAPAE. In the MVS/Batch and TSO environment this name must be LDAPAE. You can alter this name in the CICS environment as described below.

Calling SMTP-Send's LDAP lookup program from MVS/Batch or TSO

When SMTP-Send runs in MVS Batch or TSO the character 'B' is appended to the LDAP lookup program name prefix automatically when it is called by SMTP-Send. Your exit code must call the program LDAPAE.B.

Calling SMTP-Send's LDAP lookup program from CICS

When SMTP-Send runs in CICS, the character 'C' is appended to the LDAP lookup program name prefix automatically when it is called by SMTP-Send. Since your exit program is called with a CICS LINK the runtime environment cannot be shared with the SMTP-Send mainline.

Your exit code must call the SMTP-Send LDAP exit with an `EXEC CICS LINK` command. The program name for the LINK will be the LDAP program prefix with the characters 'CX' appended to it. E.g. if the `LDAP_LOOKUP_PGM` program name is LDAPAE, you should issue `EXEC CICS LINK PROGRAM(LDAPAE.CX)` with the LDAPDATA structure as described below as the COMMAREA.

Passing Data to LDAP Lookup Program

Data is passed to the LDAP lookup program (called LDAPAE hereafter) in the LDAPDATA structure. The structure allows data to be communicated between your exit program and LDAPAE. Both data used in searching and data returned are specified in the LDAPDATA structure. In addition, LDAPAE requires information about the location of your LDAP server and signon information (if required by your server). This data is passed in the LDAP parameter list string pointed to by the LDAPDATA structure as described below. If you already use LDAP for address lookups, then you can use the same data that was specified on the LDAP_LOOKUP_PGM configuration statement. A pointer to this parameter list and its length is passed to your exit. You can override the address template and the LDAP attribute from which data is extracted by coding overrides in the LDAPDATA structure.

LDAPDATA layout

The data structure LDAPDATA in INSTLIB describes the layout of the data structure passed to LDAPAE.

Parameter	Description
int[4]	Contains the return code from the user exit. This is in addition to the value returned via R15, since CICS applications are unable to return values in registers.
address[4]	Contains a pointer to a 12-byte workarea used by the LDAP routines. This value should not be altered between LDAP calls.
address[4]	Contains a pointer to the LDAP program parameter list. See below for a description of this parameter list. The parameter string specified on the LDAP_LOOKUP_PGM configuration statement and its length is passed to the validation and address mapping exits in their respective data structures and can be used to build this string.
byte[1]	Contains the length of the LDAP program parameter list (1-255 characters)
character[8]	The current jobname, substituted into %JOB template variable.
character[8]	Requesting user. RACF or equivalent or CICS userid of the user requesting the send operation. Substituted into %JUID template variable.
character[80]	Default domain specified during SMTP-Send customization, substituted into %DOMAIN variable.
byte[1]	Contains the length of a message returned by LDAP. If no message is returned this value will be set to zero. The maximum length that can be displayed by SMTP-Send is 125 characters.
character[125]	Contains a message returned from LDAP.
character[1]	If non-zero, indicates that the name of a source field is specified. Zero indicates that the default, MAIL will be used.
address[4]	pointer to field2 value, substituted into %2 template variable.
int[4]	length of field2 value, must be between 1-128.
address[4]	pointer to field1 value, substituted into %1 template variable
int[4]	length of field1 value, must be between 1-128.
character[128]	Data for substitution into %DATA template variable.
byte[1]	length of returned data
character[128]	Returned data.
byte[1]	reserved
byte[1]	x'00' or x'40' indicates V1 format, C'1' indicates V2 and presence of LDAPCOMX
byte[1]	reserved
int[4]	length of %data variable, must be between 1-128.
address[4]	optional pointer to name of the source field LDAP data will be extracted from.
int[4]	length of optional addressing template. Zero indicates no template was supplied, template in parameter list, if any, will be used.
address[4]	pointer to optional addressing template. Presence of a template overrides any template specified in the parameter list.
int[2]	time in seconds to wait for a response from a LDAP search request
int[2]	number of times to attempt to connect to the LDAP server
int[2]	time in seconds to wait between connection attempts
byte[2]	reserved
LDAPCOMX	
Start of V2 extension to LDAPDATA	
address[4]	pointer to encryption routine supplied by SMTP-Send
address[4]	pointer to decryption routine. Supplied by SMTP-Send on calls to LDAP or by user exits when calling LDAP from within an exit. Zero indicates that the password described below is in plain text. When calling encryption or decryption programs, pass the address of the following structure in General Register 1
byte	length of plain text password before encryption or after decryption.
byte	length of encrypted password after encryption or before decryption.
character[128]	plain text password value. Place value to be encrypted here before calling encryption program. Value is replaced by decryption program
byte	reserved
character[128]	encrypted text password value. Place value to be decrypted here before calling decryption program. Value is replaced by encryption program
byte	reserved

The LDAP program will return:

Return Code	Meaning
-1	no entry matched
0	normal, returned requested value in returned data
4	multiple matches found. Message area contains as many copies of returned data as will fit
8	invalid or missing parameters given
12	LDAP Bind failed
16	LDAP Open failed

By default, LDAPAE returns the value from the LDAP 'MAIL' attribute when a matching entry is found. You can change the LDAP attribute used by either a) adding the name of the field as the 6th parameter as shown below or b) passing the address of the LDAP attribute used to extract data from, in the optional LDAPDATA source field. If the LDAPDATA source field contains an address, that value will be used and any 6th parameter if present is ignored.

```
"IPAddress:389,bind user,bind pwd, ,(Hostemail=%DATA),sourceattribute"
```

With this parameter string, when searching for a matching entry for the %DATA variable "FRED", SMTP-Send uses the following LDAP search filter:

```
search base:          default
search filter:       (&(Hostemail=FRED))
attributes to return: sourceattribute
```

Examples:

```
ldap.bigfoot.com
ldap.bigfoot.com:389
ldap.bigfoot.com,binduser,bindpwd
127.0.0.1:389,"Acme Admin",password
127.0.0.1,"cn=user,cn=domain",pass,D

127.0.0.1,,D
127.0.0.1:389,,,D,otherMailbox=SNADS$%2(%1)
127.0.0.1:389,,,D,"(uid=%1)(proposedaltorgunit=%2),e_mail"
```

Searching using LDAP Lookups

UID search

In the absence of an address template passed in the LDAP Program parameter list, LDAP lookup performs a UID search. This searches for the %1 value (or the %DATA value, if %1 is not passed), in the LDAP UID attribute (**Note:** when connected to a Microsoft Active Directory LDAP server, SMTP-Send will automatically search the SAMACCOUNTNAME attribute instead of the UID attribute). The search filter will select only person entries. The search base is left to default. For example, consider a request with only the symbol %DATA defined with a value of FRED. The resulting LDAP search filter is:

```
search base:          default
search filter:       (&(objectclass=person)(uid=FRED))
attributes to return: mail
```

Template search

The template search is performed when the installation needs to map simple name values, searching either an attribute other than UID or multiple attributes. When the caller passes an addressing template, the LDAP lookup substitutes simple name values into the template, where the symbols, %1, %2, %DATA, %JOB, %JUID or %DOMAIN are specified. The resulting string is then used to build the actual LDAP search filter. The search filter will select only entries with a LDAP objectclass of person. The search base is left to default. For example, consider a request for a simple name of FRED and the following addressing template, Hostuserid=%DATA. The resulting LDAP search filter is:

```
search base:          default
search filter:       (Hostuserid=FRED)
attributes to return: mail
```

Template search using multiple attributes

A template search can be used to match on multiple attributes. Since the template is used as the LDAP search filter, it must be specified with a valid LDAP format. Each attribute=value should be enclosed within parentheses. For example, the template (Hostemail=Y)(userid=%JUID)(mail=%DATA) where the job was submitted by Sam and foo@bar.com was passed in the DATA field with a source field of location in the LDAP parm string will result in the following LDAP search filter:

```
search base:          default
search filter:       (&(Hostemail=Y)(userid=Sam)(mail=foo@bar.com))
attributes to return: location
```

A sample parm string using this template would be:

```
"IPAddress:389,bind user,bind pwd, ,(Hostemail=Y)(userid=%JUID)(mail=%DATA),location"
```

Ambiguous search results

When either a template or UID search returns more than 1 result, LDAPAE returns the ambiguous result code back to the caller. The message buffer in the caller's commarea contains as many of the resultant attribute values as will fit.

Migrating from a release prior to Release 5.2.0

Release 5.2.0 introduced some significant changes in performance to SMTP-Send's LDAP routines. To provide these improvements, the linkage used to call the SMTP-Send provided LDAP exits changed with this release. If you do not call SMTP-Send's LDAP routines from a SMTP-Send exit or provide your own LDAP routines, you can run SMTP-Send with a pre-Release 5.2.0 OPSCONFG file but you will not get any improvements when expanding distribution lists. You must specify the LDAP distribution lookup information with the LDAP_DISTLIST_PGM OPSCONFG statement to see the improvements.

The statements LDAP_LOOKUP_PGM, and LDAP_DISTLIST_PGM both cause SMTP-Send to use the 'new' style linkage, and are not compatible with a customer written exit, unless the exits are updated.

Migrating custom LDAP exits

A customer written program used to call LDAP without using routines supplied with SMTP-Send must be called using one of the following OPSCONFG statements:

```
MAPPINGPGM custom_addressing_exit_name,parms  
DISTLISTPGM custom_distribution_exit_name,parms
```

These programs will be called as separate runtime environments or LE enclaves from the SMTP-Send mainline and other than the 12 byte token passed to these programs, SMTP-Send makes no effort to retain any data or environmental state from one call to the next.

Customers calling SMTP-Send's LDAP exits from their own lookup, data or address validation exit.

Calling SMTP-Send's LDAP exit from a customer written exit in MVS/Batch or TSO

This is supported without any changes required, and with the full benefits of the enhanced LDAP exits.

In addition, the customer exit can now use a different LDAP session from the one used by SMTPSend. These LDAP session parameters are passed by the customer exit using the same interface (i.e. they are not specified to SMTPSend). Specifically, this means the customer exit can query a different LDAP server, using a different userid & password, than what is specified in the CONFIG file.

Calling SMTP-Send's LDAP exit from a customer written exit in CICS

Changes are required to call LDAP from your CICS exit and the performance improvements of the enhanced LDAP exits are not available. You must recode your exit to call the 'CX' suffixed version of LDAPAE instead of the 'C' suffixed version as in earlier releases. The calling conventions and data passed remain the same as in previous releases.

Supplementary LDAP reading

The LDAP protocol is defined by a number of Internet Engineering Task Force (IETF) request for comments (RFCs.) The IBM redbook, Understanding LDAP, (SG24-4986), provides a good introduction to LDAP, with many diagrams to assist you in understanding the concepts behind the RFC contents.

You may also need to refer to these RFCs if you customize the LDAP search filter.

- [RFC 1777 Lightweight Directory Access Protocol](#)
- [RFC 1778 The String Representation of Standard Attribute Syntaxes](#)
- [RFC 1779 A String Representation of Distinguished Names](#)
- [RFC 1960 A String Representation of LDAP Search Filters](#)

Log record format

Overview

The log records that SMTP-Send creates include standard SMF or CICS headers, followed by user data. This user data consists of three sections:

1. offset/length values for variable fields (third section)
2. fixed length fields
3. variable length fields

Offset/length values

The first section consists of a series of 8 offset/length value pairs for the variable fields. These values appear in the order FROM:, REPLY-TO:, SUBJECT:, TO:, CC:, BCC:, ATTACH and PATHINFO. The offset and length values apply to the field data only and do not include the character introducers. Offsets are halfword values that define the beginning of the field relative to the beginning of the user data area. Lengths are halfword values that define the length of the corresponding field.

Fixed length fields

The data in each field is left justified and padded with null (x'00') characters and ends with at least one space. The fixed length data area is formatted as follows:

Field	Description
VERSION[2]	X'01' Product and PATHOFF/PATHLEN information available
PRODUCT[1]	G - OPSGS3B, H - OPSGS3 (SMTP-Send Batch, SMTP-Send CICS) O - PIPGS3B, P - PIPGS3 (OP/SMTP-Send Batch, SMTP-Send CICS, OfficePath syntax) R - RAPIDGS3, S - APIGS3 (OP/SMTP-Send Batch, SMTP-Send CICS, RAPID syntax) Q - SSWDA + PIPGS3B (OP/SMTP-Send Batch for Softswitch, OfficePath syntax) T - SSWDA + RAPIDGS3 (OP/SMTP-Send Batch for Softswitch, RAPID syntax) J - OPSSSRVR (TBS SMTP Spool Server)
RSVD[3]	reserved
EXTOFFST[2]	offset to future Log record extensions
JOBNAME[9]	name of the job
JOBUSERID[9]	name of the user who submitted job
RETURNCODE[5]	job return code
TOTAL SIZE SENT[9]	number of bytes of data sent
DATE[32]	date and time sent, including GMT offset Example: Thu, 22 Nov 2001 11:51:44 -0500 Note: for items relayed by Spool Server may contain the value from the Date: header in the message.

Variable length fields

The remainder of the log record is the variable length fields. Each field is prefixed by a character introducer like FROM:, TO:, etc. Fields with multiple values are separated by commas. Only TO: and FROM: are required; the other fields are optional.

Field	Description
FROM:	FROM address or USER NAME
REPLY_TO:	REPLY_TO address
SUBJECT:	message subject (up to 256 characters)
TO:	one or more TO addresses separated by commas
CC:	one or more CC addresses separated by commas
BCC:	one or more BCC addresses separated by commas
ATTACH:	one or more attachment file names separated by commas
DATAPATH:	Information as to the delivery path used for the message: TCP(<i>ip_address</i>) - Primary or secondary SMTP relay server SPOOL(<i>node,wtrid,class</i>) - SMTP Server STC information DDName(<i>ddname</i>) - debugging redirection to external file TSQ(<i>8-char-CICS-TSQid</i>) - debugging redirection to CICS TSQ Fax(<i>Applid,TSQ: 8-char-CICS-TSQid, DDNAME:filename</i>) - send to OV/Fax V3 Unknown - message was not delivered

The following example shows 32 bytes of offset/length values, followed by 72 bytes of fixed length fields, followed by variable length fields in both the raw data format and an annotated form.

```
+0 x'006E0010',x'00890010',x'00A30016',x'00BE0010',
+10 x'00D3000D',x'00E6000F',x'00FE0014',x'011D0015',
+20 x'0001',c'G',x'000000',x'0000',c'TESTJOB '
+30 c' USERID1 0000 0'
+40 c'0001916 Tue, 24 Jun 2003 10:49:1'
+60 c'3 -0400 From: DEMO@tbsssoft.com R'
+80 c'epl-y-To: DEMO@tbsssoft.com Subjec'
+A0 c't: Demonstrate Log layout To: DE'
+C0 c'MO@tbsssoft.com Cc: cc@tbsssoft.com'
+E0 c' Bcc: bcc@tbsssoft.com Attach: TB'
+100 c'S00.OPSTEST.ASAMSG DataPath: TCP'
+120 c'(smtp.tbsssoft.com)'
```

Annotated record user data format

Offset	Value	Description
+0	x'006E0010'	From Offset x'6E', length 16 (x'10')
+4	x'00890010'	Reply-To Offset x'89', length 16 (x'10')
+8	x'00A30016'	Subject Offset x'A3', length 21 (x'16')
+C	x'00BE0010'	To Offset x'BE', length 16 (x'10')
+10	x'00D3000D'	CC Offset x'D3', length 13 (x'0D)
+14	x'00E6000F'	BCC Offset x'E6', length 15 (x'0F)
+18	x'00FE0014'	Attach Offset x'FE', length 20 (x'14')
+1C	x'011D0015'	DataPath Offset x'11D', length 21 (x'15')
+20	x'0001'	Version 1 of Log Record format
+22	c'G'	SMTP-Send Batch
+23	x'000000'	reserved
+26	x'0000'	Offset to future extensions
+28	c'TESTJOB '	Jobname
+31	c'USERID1 '	Userid
+3A	c'0000 '	completion code
+3F	c'00001916 '	total message size
+48	c'Tue, 24 Jun 2003 10:49:13 -0400 '	time and date of send
+68	c' From: '	From introducer
+6E	c'DEMO@tbsssoft.com'	From Address
+7E	c' Reply-To: '	Reply-To introducer
+89	c'DEMO@tbsssoft.com'	Reply-To address
+99	c' Subject: '	Subject introducer
+A3	c'Demonstrate Log layout'	Subject
+B9	c' To: '	To introducer
+BE	c'DEMO@tbsssoft.com '	To address
+CE	c' Cc: '	Cc introducer
+D3	c'cc1@tbsssoft.com'	CC address
+E0	c' Bcc: '	Bcc introducer
+E6	c'bcc@tbsssoft.com'	BCC address
+F5	c' Attach: '	Attachments Introducer
+FE	c'TBS00.OPSTEST.ASAMSG'	Attachment data set name
+112	c' DataPath: '	Data Path Introducer
+11D	c'TCP(smtp.tbsssoft.com)'	Data delivery path



Dynamic allocation keywords

The following are the dynamic allocation keywords that SMTP-Send recognizes:

Identifier	JCL Equivalent	Format	Description
blocklen	SPACE=(len,...)	int	Average block length for space allocation
cdisp	DISP=(s,n,cd)	KEEP CATLG DELETE	Conditional (ABEND) disposition
cyl	SPACE=(CYL,...)	none	Allocate space in cylinders
dataclas	DATACLAS=	char[9]	SMS data class
dcbddname	DCB=*	char[9]	Copy DCB information from DD statement
dcbdsname	DCB=ds	char[47]	Copy DCB information from dataset
defer	UNIT=(,DEFER)	none	Defer mounting until open
dest	DEST=	char[9]	SYSOUT destination node
destuser	DEST=(n,user)	char[9]	SYSOUT destination userid
dir	SPACE=(u,(p,s,d))	int	Number of directory blocks
disp	DISP=	NEW OLD MOD	Dataset allocation status
dsname	DSN=	char[47]	Name of dataset to allocate
expdt	LABEL=EXPDT=	char[5] char[7] ¹	Expiration date
hold	HOLD=YES	none	Hold SYSOUT for later processing
labelseq	LABEL=seq	int	File sequence number
like	LIKE=	char[47]	DSname of a model dataset (SMS)
member	DSN=ds(mem)	char[9]	PDS member name
mgmtclas	MGMTCLAS=	char[9]	SMS management class
ndisp	DISP=(s,n)	KEEP CATLG DELETE	Normal Disposition
password	none	char[9]	Dataset password
private	VOL=(PRIVATE,)	none	Private volume required
protect	PROTECT=YES	none	Protect dataset with RACF
refdd	REFDD=	char[27]	Copy DCB attributes from DDname
retention	LABEL=RETPD=	int	Retention period
rise	SPACE=(,RLSE)	none	Release unused space
round	SPACE=(,,,,ROUN D)	none	Round up space request
secmodel	SECMODEL=	char[47]	SMS security model dataset (2)
space1	SPACE=(u,(s1,s2))	int	Primary space allocation
space2	SPACE=(u,(s1,s2))	int	Secondary space allocation
storclas	STORCLAS=	char[9]	SMS storage class
sysout	SYSOUT=	char[1]	Sysout class
sysoutform	SYSOUT=(c,,f)	char[5]	SYSOUT forms
sysoutwtr	SYSOUT=(c,p)	char[9]	SYSOUT writer program name
tracks	SPACE=(TRK,...)	none	Allocate space in tracks
volcount	VOL=(,,,count)	int	Maximum number of volumes
volref	VOL=REF=	char[46]	Request same volume as another dataset
volseq	VOL=(,seq)	int	Sequence number of first volume to mount
volume	VOL=SER=	char[7]	Multiple Names of required volumes

¹ An expiration date may be specified in one of three formats. In a five-digit expiration date, the first two digits are interpreted as the years since 1900. In a seven-digit expiration date, the first four digits are the entire year. The format "yyyy/ddd", as used with the JCL EXPDT keyword, is also supported.

OfficePath CARDFILE control statements

OP/SS The table summarizes the OfficePath and RAPID control statements processed by OP/SS. The rest of the OfficePath and RAPID control statements have no equivalent in SMTP and so 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]	<ol style="list-style-type: none">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]	

Change history

This is a summary of significant changes to SMTP-Send and OfficePath/SMTP-Send.

R5.2

Enhancements

- SMTP Leading dot Transparency feature added to TBS Spool Server
- RemoveTrailingBlanks feature added to OPSCONFG.
- Performance improvements to the LDAP exits add new OPSCONFG parm LDAP_DISTLIST_PGM, which is used to call the new LDAP exits, while the old parm, Distlistpgm, maintains the old calling style.

Fixes

OPS144 - Abend S0C4 in OPSGS3B after 2 consecutive address resolution failures
OPS143-OPSSSRVR Socket Receive error when data contains leading period
OPS142-Default text encoding is 7bit. 8bit required by ISP
OPS141-Text of message does not appear in message sent to cell phone
OPS140-PDF output is being truncated
OPS139-When OPSSSRVR in retry wait and P OPSSSRVR issued, current message is deleted from JES Spool
OPS138-OPSSSRVR GETMAIN failure in TSSSEND
OPS137-Endless loop checking response from server
OPS136-Abend S0C4 when default Papersize not specified in CONFIG
OPS135-PDF (LPP(xx)) not recognized with ATTACH keyword
OPS134-Attachment converted to PDF corrupted if redirected to spool when both servers are unavailable
OPS133-Msg LSCX701 ENFOUND loading module OPSCFDSN
OPS132-OPSSSRVR - Job disappears from Spool when both open of both Primary and Secondary fail
OPS131-Abend S0C4 sending large PDF attachment

R5.1

Enhancements

- New OUTTYPE keyword, MAPHTML causes any '<', '&' or '>' characters in the input MESSAGE stream to be mapped into < & and > strings. This will prevent their being interpreted as part of HTML tags. Default is to pass embedded HTML characters through. You can also surround a HTML tag with ! characters, i.e. !! to ensure that the string will not be interpreted as HTML. The tag and the surrounding '! characters must not continue across a line.
- OPSCONFG record size increased to allow 256 characters of data instead of 128.
- Customized EBCDIC/ASCII translation table can now be dynamically loaded. Code the OPSCONFG control statement. UseCustomCodepage YES, to force SMTP-Send to attempt to load the load module, XLTnnn where nnn is the value specified on a CODEPAGE control statement. You create the customized translation table using the sample XLT999 and job OPSI12.
- New OPSCONFG statement, FAIL_ON_VALIDATION_ERROR added to highlight address validation errors. When FAIL_ON_VALIDATION_ERROR YES is coded, and an address fails validation, SMTP-Send will terminate with RC=32
- New INSTLIB member, OPSI13 will write the current version of SMTP-Send, the Configuration program, the Spool Server and SSWDA to the joblog for easy determination of which release is running.
- New CICS commarea structure allows the separation of control statements from data in two separate TSQs. In addition, simple sends are possible by specifying the To, From and Subject information directly in the commarea.
- New program link interface allows for sending of single emails without having to create MESSAGE and or CONTROL datasets. The envelope information and message text are supplied in buffers passed from the calling program using the OPSBCALL data structure.
- SENDEEMPTYATTACHMENTS YES|NO OPSCONFG statement added to control the sending of empty attachments. You can also code OUTTYPE EMPTY_ATT or NOEMPTY_ATT to control this behavior at the distribution level.
- Spool Server sends undeliverable messages as non-delivery reports to the user specified in the ERROR_MSG_TO_NAME

Improvements

- TBS Spool Server performance has been substantially improved.

Fixes

OPS130 - LSCX110 error in MEMCPY called from line 557 of @@650343(TSSSEND)
OPS129 - Extraneous JOB info on HELO Statement rejected by receiving SMTP server
OPS128 - Error in reported size when SMTP-Send rejects message with attachment as too large
OPS126 - Bypass sending attachment if it's empty
OPS125 - xxxGS3 ending with RC 105 When SEND_TEXT=SPOOL
OPS124 - xxxGS3 EXEC CICS FAIL: EIBFN: 0a, EIBRESP: 44, EIBRESP2: 0 when TSQ name contains NULL character
OPS123 - SYSTEM COMPLETION CODE=66D REASON CODE=00000031 when temp file allocation fails
OPS122 - Throughput of OPSSSRVR needs improvement
OPS121 - Faxes sent by OP/SS-Fax arrive empty in OV/Fax
OPS120 - RC=514 while testing SMTP-Send with Invalid recipient should be RC=504
OPS119 - Imbedded HTML tags no longer working with R5.0.6
OPS118 - MAILTO: prefix of email address not a valid hotspot
OPS117 - CPU loop in GSRCHRP@ when multiple emails passed in MESSAGE input

OPS116 - LOOKUPER file remains empty
 OPS115 - CICS TSQ from commarea truncated to 7 byte length
 OPS114 - LSCX130 msg and U1208 error with SUBJECT >80
 OPS113 - Empty file gets condition code of zero instead of four
 OPS112 - RC=505 issued instead of RC=08 when validation exit denies address
 OPS111 - MSG "IKJ56241I REQUEST REQUIRES MORE NON-SMS MANAGED VOLUMES THAN ARE ELIGIBLE" with tempfile input
 OPS110 - Incorrect translation for Canadian French Characters
 OPS109 - Msg "File not found" when MESSAGE file is empty
 OPS108 - LDAP Return attribute field name ignored after R5.0 install
 OPS106 - Adobe cannot READ PDF zipped file after redirection to JES spool
 OPS105 - Encrypted passwords available in clear text in dump
 OPS104 - Msg "No valid Sender or ReplyTo address" when unbalanced parens in AUTHOR statement
 OPS103 - '501 Failure Naked LF' message returned by receiving SMTP server
 OPS102 - Abend S0C4 in TSSJES when library/module is NOT APF-authorized

R5.0

Enhancements

- LDAP lookups enhanced:
- For address lookups, you now have complete control over the OBJECTCLASS= value for address lookups when you pass an address template. SMTP-Send will no longer add "(OBJECTCLASS=person)" to the beginning of any address template passed (as was done in previous releases). If you need to preserve this behavior of previous releases you will have to add this string to the start of your template. Removing this default allows the address lookup to locate OBJECTCLASS=GROUP entries that have MAIL= attributes. Note: if your LDAP parm string never contained a template, the default template (&(OBJECTCLASS=person)(UID=%DEN)) will continue to be used.
- For distribution list lookups, all searches for group entries will now use the value of a MAIL= attribute on the entry, if one is found. This is added to the list of any MEMBER= values that have always been obtained from group entries.
- If the LDAP lookup fails to connect successfully with the specified LDAP server, SMTP-Send will now terminate with condition code of 32.
- LDAPDL group lookups now use the MAIL= attribute if available.
- Support for encrypted LDAP passwords is now available. OPSCONFIG statements LDAP_Password value and Exit_Password value are added to supply passwords that will be encrypted in the VSAM Config file LDAP_Password is used to specify passwords used in the LDAP_LOOKUP_PGM exit program. Exit_Password supplies the password used in the MAPPINGPGM exit. For the DISTLISTPGM exit program, the LDAP_Password value is used when the exit program name begins with LDAP and the Exit_Password value is used otherwise.
- You indicate that an encrypted password is to be used by specifying an asterisk as the password in the program parameters, i.e. LDAP_LOOKUP_PGM 'LDAPAE,IP 127.0.0.1,User,*'
- You can use encrypted passwords in your own calls to LDAP by specifying an asterisk as the password value in the parameter string and supplying an encrypted password value and the address of the decryption code in the call to the SMTP-Send supplied LDAP routines. The SMTP-Send supplied encryption routine is bidirectional and you can use it if you wish to encode your password before calling LDAP. The data structures passed to the LDAP routines have been enhanced to provide this additional information.

N.B. Passwords coded in the exit specification statements will not be encrypted when added to the VSAM file and will continue to appear in plain text in SMTP-Send memory and debugging and tracing messages.

- Support added to split Text HFS files at LF characters. OPSCONFG statement ReplaceLFWCRLF {Y|N} and CONTROL - statement LF2CRLF {Y|N} added to control this support
- Additional OPSCONFG configuration statements added to provide more information about failures in sending messages via TCP/IP:
- SetTCPFailureRC {4 | 0-9999} - allows the customization of the cond code returned when SMTP-Send is unable to send a message via TCP/ IP and the message has instead been written to spool
- WTO_onTCPFailure {No | Yes} - Yes will cause the message '+OPS126E FAILURE RELAYING TCPIP DATA' to be written to the job log and system log
- Additional Control Statements now supported to indicate information about the content of the message:
 - Importance {Low | Normal | High} - indicates the importance of the message
 - Priority {Normal | Urgent | Non-urgent} - indicates the priority of the message
 - Sensitivity {Personal | Private | Confidential | Company-Confidential} - indicates the sensitivity of the message.
- Optional Sensitivity text may be added to either the top or bottom of the message when the Sensitivity control statement is coded. The addition of the text is controlled by the OPSCONFG statement, Sensitivity_Msg {None | Top | Bottom}
- The actual text is controlled through the SENSITIVITY_TXT OPSCONFG statements
- Allow Specification of HTML foreground and background colors and font face and size via Cascading Style Sheets (CSS)
 - Use of CSS is controlled by the OPSCONFG statement Use_CSS_Styles {Yes | No}
 - Style characteristics are controlled by the following OPSCONFG statements
 - BODY_Font-Size {pointsize | {xx-small|x-small|small|medium|large|x-large|x-large|xx-large}}
 - BODY_Font-Family {font [,font...]}
 - BODY_Color background_color[,foreground_color]
 - and the following Control Statements:
 - BODYCOLOR {background_color[White][,foreground_color[Black]}
 - FONTFAMILY {font [,font...]}
 - CSSSTYLE {Yes | No}.

Notes: the font size is controlled by the PITCH statement available in previous SMTP-Send releases

Input with `INTYPE TEMPLATE` is assumed to already be formatted with the required HTML and therefore CSS styles will not be added by default. Specifying an `OUTTYPE` of `HTML` or `HTMLPRE` with `TEMPLATE` input will force the addition of CSS style information if available and requested by the `Use_CSS_Styles` statement

- A started task program, `OPSSSRVR`, is now included to relay any SMTP data resident on the JES Spool to a relay server. SMTP data may be placed on spool by SMTP-Send for safe storage when it is unable to access either of the specified Relay servers or by other utilities such as IBM's SMTPNOTE REXX exec. `OPSSSRVR` offers the same e-mail traffic logging mechanisms as native SMTP-Send.
- Installations running the IBM supplied SMTP Server task to relay messages from the JES Spool can replace the IBM task with `OPSSSRVR`. `OPSSSRVR` does not listen for SMTP traffic over TCP/IP, only accepting SMTP mail from the JES Spool and therefore cannot be used as an open relay for unsolicited e-mail traffic.
- Spaces may now be specified in file names in the `FILENAME` and `DOCNAME` statements and for the `newname` field in the `ATTACH` statement. Enclose the name containing spaces in double quote marks.
- Symbolic substitution in message text, subject and filenames. Certain date symbols (where noted) may include a offset (positive or negative) to be used in calculating the displayed date from the current date.
- This functionality is active by default but may be turned off for a specific send with the `NOSYMBOLS` keyword of the `OUTTYPE` Control statement
- The TSO/ISPF dialogs have been enhanced to allow requesting that a file sent as an attachment be translated from EBCDIC to ASCII. Previous release always sent attachments as binary objects.
- A sample REXX Exec, `OPSREXX` is now available in `INSTRIB` to demonstrate calling SMTP-Send from REXX
- Files attached using the `ATTACH` control statement can now be converted to PDF format.
- Additional control over conversion to PDF added via `PDFxxxx` control statements and `PDF` subparameters on `ATTACH`.
- Multiple keywords now allowed on `OUTTYPE` statement when `ATTACHMENT` is specified. For example, you can now request that the MESSAGE dataset be sent as an attachment and be enveloped in HTML headers by specifying `OUTTYPE ATTACHMENT HTMLPRE`
- Logging updated to include the data path the message took, i.e. TCP or SPOOL.
- e.g. `DataPath: Spool(N1.SMTP,B)` or `DataPath: TCP(smtp.tbssoft.com)`
- `OPSNOTIFY` parms updated.
- `Create_MailTo_Links` `OPSCONFG` statement added
- `NOMLINKS` Control statement added
- For RTF attachments, SMTP-Send now dynamically calculates the appropriate papersize to contain the data. You can override this calculation by specifying the `PAPERSIZE` control statement. The parameters are the same as those used by the `PDFPAPER` control statement
- `OPSCONFG` statement `UseDLAddress {Yes|No}` added as part of the fix for OPS-089 to control the expansion of Distribution lists during LDAP processing. By default, SMTP-Send will now not expand a Distribution List with an email address associated with it, but instead will merely send the email to that address, letting the receiving email system expand the address. This is a change from previous behavior which can be restored by specifying `UseDLAddress No`
- A Customized EBCDIC/ASCII translation table can now be dynamically loaded. Code the `OPSCONFG` control statement. `UseCustomCodepage Yes`, to force SMTP-Send to attempt to load the module, `XLTnnn` where `nnn` is the value specified on a `CODEPAGE` control statement. You create the customized translation table using the sample `XLT999` and job `OPS112` in `INSTRIB`.

Miscellaneous Documentation Updates

- LDAP userid and password fields containing spaces must be enclosed in double quotes, forcing the entire LDAP statement containing the double quotes to be enclosed in single quotes.
- Updated Return Code values
 - 6 - no longer used
 - 32 - returned upon LDAP communication failure
 - 1046 - `OPSCONFG` file cannot be found
- IBMMAIL format input cannot be specified in HFS files
- `OPSSSRVR` - SMTP relay server documentation now included

Fixes

OPS101 - '@' sign causes incorrect mailto: HTML tag to be generated
OPS100 - LDAP_PASSWORD parms in config not working, Abend S0C4 results
OPS099 - Inconsistent output with Attachments using RTF tags
OPS098 - CONFIG job abends S0C1 - OPSENCPT module missing
OPS097 - Incorrect attachment name if new name starts with 'TR'
OPS096 - PIPGS3B malloc() failure, size 00000 when CARDFILE=DD DUMMY
OPS095 - Abend S0C4 in valRecipientName if FROM is not found and lookup_RACFID_as_Sender=yes
OPS094 - Cannot create an attachment starting with 'TR'
OPS093 - Dropping carriage control plus other characters on first line of attachment when converting to RTF
OPS092 - PDF files arrive in binary not as an attachment when SEND_TEXT SPOOL specified in OPSCONFG
OPS091 - Multiple Send from CICS not working as end boundary not recognized
OPS090 - OPSGS3B Return code 01030 when trailing boundary encountered
OPS089 - SMTP-Send not delivering to addresses in Exchange DL's with Hide-From-Address-Book=1
OPS088 - Allocation failure when creating PDF attachments. LSCX862
OPS087 - LSCX111 message sending PDF attachments
OPS086 - Low value (NULL) chars in MESSAGE input causes incomplete mail to be sent
OPS085 - SENDMERGE fails if CC and BCC not also specified as global or local
OPS084 - RAPIDGS3 SUBPARAMETER INVALID FOR THIS REQUEST: ILLEGAL DCALEVELVALUE = 01403
OPS082 - Sensitivity message at top when Sensitivity_Msg=Bottom
OPS081 - Multiple sends via TCPIP fail with failure to open LISTDATA
OPS080 - Attachment not zipped after R5.0.1 installation
OPS079 - OPSGS3B 0505 error - code not appending domain name
OPS078 - Email is blank after R5.0 when SEND_TEXT Spool specified
OPS076 - Attachments from ATTACH statement not attached when MESSAGE input sent as attachment
OPS075 - No message returned to sender if Receiving SMTP server fails the send

OPS074 - 'Empty Recipient List' messages returned when relay server rejects the message
OPS073 - Proposed SMTP Name missing from validation exit call
OPS071 - OPSGS3 fails to read MESSAGE input when writing to spool
OPS070 - Data truncated in display of PDF file wider than 80 characters
OPS069 - SMTP-SEND ZIP function did not zip PDS. Error msg not issued
OPS068 - Zip file attachment is corrupted after redirect of TCP data to JES Spool
OPS067 - LDAP times out immediately with timeout of 0 or after upgrade to 4.2.13 from release 3
OPS066 - Carriage Controls not handled properly for Zipped Attachments from ATTACH statements
OPS065 - ZIP of Translated Text Attachment is unreadable
OPS064 - RC = 0 when some userids fail to map to SNADS userids to email addresses
OPS063 - No way to specify a new name for an attachment that contains a space
OPS062 - PIPGS3B LRECL 137 TOO SHORT FOR FILE DATAOUT, 138 REQUIRED
OPS061 - Multiple sends via MESSAGEDD only (PARM=P) fail after first send with ENOTOPEN error
OPS060 - LDAP Passwords exposed in VSAM file and debug traces
OPS059 - Unable to attach HFS file

R4.2

- NOTICE, SENDER, MSG_IN_XMESSAGEF, USE_MIME_FOR_1403, new CONFIG control statements
- ATTACHMENT, NOMIME, RTF keywords, OUTTYPE statement
- TRANSLATE, RTF keywords, ATTACH statement
- Logging to SMF, CICS journal or sequential file

1. The previous releases were SMTP-Send R2.1 and OfficePath/SMTP R4.1. The release numbers for both products were synchronized starting with R4.2.