

---

# XMITIP User Reference Guide

## *SMTP (E-Mail) from z/OS to the World*

Version 18.03

Revised March 28, 2018

Lionel B. Dyck  
E-Mail: [lbdyck@gmail.com](mailto:lbdyck@gmail.com)

## Table of Contents

|   |           |
|---|-----------|
| <b>Table of Contents</b> .....  | <b>2</b>  |
| <b>Introduction</b> .....   | <b>5</b>  |
| <b>Acknowledgements</b> .....   | <b>6</b>  |
| <b>Local Customizations (alter this section for your site and republish for your users)</b> ..... | <b>7</b>  |
| <b>Using XMITIP in Batch</b> .....  | <b>8</b>  |
| <b>Usage Notes</b> .....  | <b>9</b>  |
| Performance Considerations .....  | 9         |
| <b>XMITIP - The Command (syntax)</b> .....  | <b>10</b> |
| To-Address.....   | 11        |
| Other Keywords.....   | 11        |
| AddressFile and AddressFileDD .....   | 11        |
| ASA .....   | 12        |
| BCC .....   | 12        |
| CC.....   | 12        |
| CONFIG .....  | 12        |
| CONFIGDD .....  | 13        |
| DEBUG .....   | 13        |
| EMSG .....  | 13        |
| ERRORSTO .....  | 13        |
| FILE .....  | 13        |
| FILEDD .....  | 13        |
| FILEDESC.....   | 14        |
| FILENAME .....  | 14        |
| FILEO.....  | 14        |
| FOLLOWUP .....  | 14        |
| FORMAT.....   | 15        |
| FORMAT TXT.....   | 15        |
| FORMAT CSV .....  | 15        |
| FORMAT BIN .....  | 15        |
| FORMAT GIF .....  | 15        |
| FORMAT ICAL.....  | 15        |
| FORMAT HTML .....   | 15        |
| FORMAT PDF .....  | 16        |
| FORMAT RTF.....   | 17        |
| FORMAT XMIT .....   | 18        |
| FORMAT ZIP.....   | 18        |
| FORMAT ZIPBIN .....   | 18        |
| FORMAT ZIPCSV.....  | 18        |
| FORMAT ZIPGIF .....   | 19        |
| FORMAT ZIPHTML .....  | 19        |
| FORMAT ZIPPDF .....   | 19        |
| FORMAT ZIPRTF .....   | 19        |
| FORMAT ZIPXMIT .....  | 20        |
| FROM .....  | 20        |
| HLQ.....  | 20        |
| HTML .....  | 20        |
| IDVAL .....   | 20        |
| IMPORTANCE.....   | 21        |

|                               |    |
|-------------------------------|----|
| IGNORECC.....                 | 21 |
| IGNOREENC.....                | 21 |
| IGNORESUFFIX.....             | 21 |
| LANG.....                     | 21 |
| MACH.....                     | 21 |
| MARGIN.....                   | 21 |
| MSGDD.....                    | 21 |
| MSGDS.....                    | 22 |
| MSGQ.....                     | 22 |
| MSGT.....                     | 22 |
| MSG72.....                    | 22 |
| MURPHY.....                   | 22 |
| NOCONFIRM.....                | 22 |
| NOEMPTY.....                  | 23 |
| NOIDVAL.....                  | 23 |
| NOMSG.....                    | 23 |
| NORTFXLATE.....               | 23 |
| NOSPOOF.....                  | 23 |
| NOSTRIP.....                  | 23 |
| PAGE.....                     | 23 |
| PDFIDX.....                   | 24 |
| PRIORITY.....                 | 24 |
| RCO.....                      | 24 |
| RECEIPT.....                  | 24 |
| REPLYTO.....                  | 24 |
| RESPOND.....                  | 24 |
| SENSITIVITY.....              | 25 |
| SIG.....                      | 25 |
| SIGDD.....                    | 25 |
| SUBJECT.....                  | 25 |
| TLS.....                      | 26 |
| TPAGELEN.....                 | 26 |
| VIANJE.....                   | 26 |
| ZIPMETHOD.....                | 26 |
| ZIPPASS.....                  | 26 |
| Minimum required syntax:..... | 28 |

**XMITIP Examples ..... 29**

|  |    |
|--|----|
| <b>Example 1:</b> Send a PDS Member with No Message.....                               | 29 |
| <b>Example 2:</b> Send a message to One Address.....                                   | 29 |
| <b>Example 3:</b> Send a Message with CC and FROM specified.....                       | 30 |
| <b>Example 4:</b> Send a Single PDS Member To One Address.....                         | 30 |
| <b>Example 5:</b> Send a Dataset in Landscape with a 9 point font.....                 | 30 |
| <b>Example 6:</b> Send a Passed Dataset to a List using Blind Copies.....              | 30 |
| <b>Example 7:</b> Send a Dataset Using an AddressFile Dataset:.....                    | 31 |
| <b>Example 8:</b> Using MSGQ.....  | 31 |
| <b>Example 9:</b> Send a SYSOUT file as a Text Attachment:.....                        | 32 |
| <b>Example 10:</b> Send a Dataset in RTF Format using ZIP to save space.....           | 32 |
| <b>Example 11:</b> Send a Comma Separated Value (CSV) Dataset.....                     | 33 |
| <b>Example 12:</b> Sample using MSGT.....  | 33 |
| <b>Example 13:</b> Using a XMITIP Configuration File.....                              | 34 |
| <b>Example 14:</b> Sending a Message to a Pager if Job Abnormally Ends.....            | 34 |
| <b>Example 15:</b> Splitting a Report into Individual E-Mails based on a KeyValue..... | 35 |
| <b>Example 16:</b> PDF creation example using a PDF Configuration File.....            | 36 |

|   |           |
|---|-----------|
| <b>XMITIPTD – Timed Delivery Routine .....</b>            | <b>37</b> |
| <b>XMITIPSP – File Split/Separation Utility.....</b>      | <b>38</b> |
| XMITIPSP Syntax.....                                      | 38        |
| XMITIPSP Control Statements .....                         | 38        |
| COMBINE .....   | 38        |
| FILENAME .....  | 38        |
| GMAIL.....  | 39        |
| KEYMAIL .....   | 39        |
| KEYPREF .....   | 39        |
| KEYV.....   | 39        |
| KEYSUBJ .....   | 39        |
| MERGEMAIL .....   | 40        |
| MSGDD .....   | 40        |
| MSGDS.....  | 40        |
| MSGSTART .....  | 40        |
| MSGEND .....  | 40        |
| SEPLINES .....  | 40        |
| SEPLLOC.....  | 40        |
| SEPPAGES .....  | 40        |
| SUBJECT .....   | 41        |
| TO .....  | 41        |
| XMITIPCONFIG.....   | 41        |
| XMITIPSP Required Statements .....                        | 41        |
| XMITIPSP Mutually Exclusive Statements .....              | 41        |
| XMITIPSP Required JCL .....                               | 41        |
| <b>ISPF Dialog.....</b>                                   | <b>42</b> |
| Batch JCL and XMITIP Generation from the ISPF Dialog..... | 44        |
| Address List Dialog .....                                 | 46        |
| File Attachment Panel .....                               | 47        |
| Format Prompting .....                                    | 47        |
| <b>XMITIP Tools.....</b>                                  | <b>49</b> |
| ALLOCGDG .....  | 49        |
| EDIMAIL .....   | 49        |
| MAILFILE .....  | 49        |
| MAILHFSE .....  | 49        |
| SETSDSFK and XMITSDSF .....                               | 50        |
| TXT2HTML .....  | 51        |
| TXT2RTF .....   | 51        |
| TXT2PDF .....   | 52        |
| XMITIPED .....  | 52        |
| XMITIPFE.....   | 52        |
| XMITIPPD.....   | 52        |
| XMITBULK.....   | 52        |
| <b>Reading the Attachment on the Workstation: .....</b>   | <b>53</b> |
| <b>Change History .....</b>                               | <b>54</b> |

## Introduction

XMITIP is a mainframe based electronic mail application that is capable of sending electronic mail to any valid Intranet or Internet address. Along with messages, XMITIP can also send mainframe files in one of several different file attachment formats. The recipients can be on any mail system that connects to the Internet (the world) or Intranet (in house). The Simple Mail Transport Protocol, or SMTP, is used for sending the mail with data sets attached using the appropriate SMTP statements.

The name, XMITIP, derives from the TSO TRANSMIT command, which has an alias of XMIT, and from IP, which stands for Internet Protocol. Thus XMITIP is TSO TRANSMIT using the Internet Protocol.

XMITIP can be executed as a step within a batch job, under TSO as a command, or under ISPF using a robust ISPF interface. It can also be used within an automated operations tool to generate messages related to system events.

XMITIP can be used very effectively as a step within a batch job to send a report generated by that job via electronic mail rather than printing the report. With this approach the report will arrive in the intended users electronic mail inbox within a few minutes after the job completes. This is much faster than waiting for the hours for the report to be printed, removed from the printer, separated from the other reports that were printed at the same time, and then delivered or placed in a mail box for pickup. The user can then view the report online or print as their needs dictate.

Some of the features of XMITIP are:

- send electronic mail to one or more addresses
- send a quick message as a page
- send one or more data sets as file attachments in one of the following formats: plain text, HTML, Rich Text Format (RTF), Portable Document Format (PDF), Comma Separated Value (CSV), TSO Transmit (XMIT), and binary
- supports address lists
- supports CC and BCC
- supports Priority, Sensitivity, and Importance
- The ISPF interface:
  - field level help for all entry fields
  - validation of all entered fields
  - address table for lookup and selection
  - data set table for multiple data set selection and formatting
  - reports the complete XMITIP command syntax generated
  - option to create a file with a complete Batch Job which can be submitted, browsed, edited, or copied.

The ISPF interface is an excellent tool to use to model the XMITIP usage and once it is working completely then the generated command can be integrated into a batch job step or TSO application.

Included with the XMITIP package are several tools that can be used to invoke XMITIP from other applications. These tools interface to XMITIP from SDSF, from IOF, and from other ISPF applications.

XMITIP is written almost completely in z/OS REXX, with one z/OS Assembler program that performs the MIME (Multipurpose Internet Mail Encoding) used for PDF and Binary attachments. The ISPF interface is also written in z/OS REXX using the z/OS ISPF APIs.

This is an evolving application and any comments, suggestions, or bugs should be reported to the author's e-mail ([lbdyck@gmail.com](mailto:lbdyck@gmail.com)). Please indicate the version of XMITIP that you are using in any e-mail.

## Acknowledgements

I would like to acknowledge the contributions of the following individuals who provided pieces of the XMITIP application that you are now installing:

**IBM** for significant assistance in answering various SMTP issues over the years.

**Dana Mitchell** for the Interlink support information.

**Doug Adams** for many ISPF table-coding examples, including the excellent FIND routines.

**Doug Nadel** for the routine to convert a number to include commas.

**Felipe Cvitanich** for his contributions of the national language enablement tools.

**Joel Ewing** for his contribution to the XMITIPSP code.

**John Ellis** for his contributions, including the interface to UDSMTP and the code to process all, or selected via mask, members of a PDS.

**Leland Lucius** for his contributions of the PDF conversion routine, the initial routine to do MIME conversion, and the time zone detection routine. Most recently for the PDF security routines and enhancements to the TXT2PDF code.

**L. A. Thomas** for taking the time to provide extensive suggestions on documents, the ISPF help panels, the ISPF dialog, and the processing of the application.

**Mark Regan** for setting up the XMITIP Listserv group on Yahoo.

**Mark Feldman** for the XMITB64 assembler program.

**Mike Porter** for the UDSMTP program.

**Ken Tomiak** and **Barry Gilder** for their contribution of the REXX routine CONDCODE that is used to capture the step completion information for the active job.

**Paul Wells** for providing the Murphy (XMITIPMU) REXX routine.

**Rich Stuemke** for the Machine Carriage Control information.

**Wolfram Schwenzler** for code to support the characters \{ } in the RTF files.

**Hartmut Beckmann** for numerous enhancements and suggestions.

**Alain Janssens** for the code in TXT2HTML to convert CSV files to HTML tables

**Yvon Roy** for code to detect daylight savings time.

Plus numerous individuals who have performed the task of beta testing as well as others who took the time to send me comments, suggestions, and bug reports.

## Local Customizations *(alter this section for your site and republish for your users)*

Some of the local customizations that you need to be aware of are:

1. ZIP option

The ZIP option is enabled and utilizes PKZIP/MVS. This utility is only available on a few of the systems. Before relying upon this option you should verify that the ZIP utility is available on the systems upon which your jobs will be executing.

2. PKZIP/MVS is not licensed on all systems. Thus if using a ZIP option in XMITIP the following JCL statement **must** be inserted into the batch JCL after the JOB statement:

```
//JBS BIND PKZIP.LIC
```

3. The Data Set used in batch mode on the //SYSEXEC DD depends upon the environment:

| Environment         | Data Set Name     |
|---------------------|-------------------|
| Hawaii              | SYS2.USER.CMDPROC |
| National            | SYS1.USER.REXX    |
| Northern California | SYS2.REXX.EXEC    |

## Using XMITIP in Batch

When using XMITIP in batch using the Batch TMP some suggestions and restrictions to be aware of are:

1. The data set name format is the same as under TSO. Fully qualify with single quotes or be aware of what your TSO PROFILE PREFIX is set to as the prefix will be appended to the start of the data set name if it is not fully qualified.
2. The SYSTSIN DD will ignore any text in columns 73 to 80.
3. Statements in the XMITIP command can be continued to additional statements by coding either a plus (+) or a minus (-) at the end of the statement to be continued. A plus (+) is recommended as it pass less white space within the merged command.
4. Messages are reported in the SYSTSPRT DD



## Usage Notes

Some considerations for using XMITIP are:

1. Use a SUBJECT to inform the recipient what the e-mail is about
2. Always use a FROM address which should map to your normal e-mail address on your production e-mail system. This allows the recipient to easily reply to your e-mail address. If you forget a FROM then a default FROM is generated using the e-mail address of the TSO session or the userid of the Batch JOB.
3. The largest record length supported is either 998 or 1024. The SMTP standards (RFC 2821) define a limit of 998 bytes per record while the IBM SMTP limit is 1024 bytes. There is an installation customization option for XMITIP that defines which limit is enforced.
4. Before implementing XMITIP into a production process run several tests to verify that the results (e.g. the report format) are what you expect.

Use the ISPF dialog to experiment with the various XMITIP options and use the generated XMITIP JCL and command as a starting point for a production implementation.

### Performance Considerations

There is overhead to using this utility however if the usage is infrequent then that overhead is normally acceptable. One of the reasons the DEBUG keyword was added to the tool was to provide information on what the SMTP data stream consists of so that a programmer can take that and implement it in their program for their specific purposes.

## XMITIP - The Command (syntax)

XMITIP is a normal TSO command and must be executed under TSO or using the batch terminal monitor program (TMP). See the Examples section for examples of the JCL which can be used, the complete syntax is:

```
%XMITIP user@address
or (u1@address u2@address..)
or *list-id*
or *
AddressFile
AddressFileDD
ASA or MACH or IGNORECC
BCC user@address
or BCC (u1@add1 u2@add2 ..)
CC user@address
or CC (u1@add1 u2@add2 ..)
CONFIG filename
or CONFIGDD ddname
DEBUG
EMSG
ERRROSTO user@address
FILE dsn
or FILE (dsn1 dsn2 ...)
FILEDD ddnl
or FILEDD (ddnl ddn2 ...)
FILEDESC file-description
or FILEDESC (desc1 desc2 ...)
FILENAME filename
or FILENAME (file1 file2 ..)
FILEO hfs-file-name
or FILEO (hfs-file-name1 hfs-file-name2 ..)
FOLLOWUP date
FORMAT options (see the doc)
FROM from@address
HLQ high-level-qualifier
HTML (for MSGDx only)
IDVAL
IGNORESUFFIX
IMPORTANCE High Normal or Low
LANG default_language
MARGIN Lm/Rm/Tm/Bm
or MARGIN (Lm/Rm/Tm/Bm Lm/Rm/Tm/Bm ...)
MSGDS data-set-name
or MSGDS *
or MSGDD ddname or MSGQ or MSGT
or NOMSG (if no MSGDS, MSGDD, MSGT or MSGQ)
MSG72
Murphy
NOConfirm
NoRTFXlate
NoSpoof
NOSTrip
PAGE `page message text`
PDFIDX row/column/length
PRIORITY Urgent Normal or Non-Urgent
RC0
RECEIPT receipt@address
REPLYTO reply@address
RESPOND option(s)
SENSITIVITY Private Personal or Confidential
SIG dsname or SIGDD ddname
SUBJECT 'subject text'
TLS On or Off
VIANJE `NJE-Node`
ZIPMETHOD zip-compression-method
ZIPPASS password for zip files
```

The command syntax is keyword or keyword followed by an option with no intervening equal sign. If multiple options are used then they are enclosed within parentheses.

**%XMITIP** is the command. The % indicates that the command is a REXX Exec and not a compiled command. If entered without any parameters either an error message will be generated or, if executed under ISPF it will invoke the ISPF interface. The command requires a minimum set of parameters to function, including a **to-address** and either a message keyword (**msgds**, **msgdd**, or **msgq**) or a file attachment (**file**, **filedd**, or **fileo**). Use of **msg72** will limit the message text to the first 72 columns (to avoid sequence numbers).

### To-Address

The only positional parameter to XMITIP is the To-Address, which is the electronic mail address of the intended recipient. Note that you do not code To-Address rather you code the actual address of the recipient. The format of this parameter is:

- address  
e.g. first.last@company.com
- "name" <address>  
e.g. "First Last" <first.last@company.com>
- (address1 address2 ...)  
multiple addresses are enclosed with parentheses separated by at least one blank or a comma
- ("name1" <address1> "name2" <address2> ...)  
multiple addresses are enclosed with parentheses separated by at least one blank or a comma
- \*  
an asterisk is used if you are using the **ADDRESSFILE** or **ADDRESSFILEDD** keywords to define a distribution list of addresses. The single asterisk will result in the **To** field being blank on the users e-mail.
- \*address-list-description\*  
this is similar to the usage of a single asterisk except that the text enclosed by the two asterisks will be specified on in the **To** field and will appear in the users e-mail. There must be no blanks between the two asterisks.

The name within quotes, referred to in this document as the qualified address, will appear in the users inbox rather than the e-mail address.

### Other Keywords

The remaining keywords used by XMITIP are not positional or required. The keywords will be presented in alphabetical order below.

### AddressFile and AddressFileDD

These keywords describe a sequential data set or member of a partitioned data set that contains a distribution list of addresses. The syntax is:

ADDRESSFILE data-set-name  
Or  
ADDRESSFILEDD ddname

AddressFile may be abbreviated as AFile.

AddressFileDD may be abbreviated as AFileDD.

The format of the address list is:

Column 1:           \* for a comment or an action (To, CC, BCC, FROM or REPLYTO)

Started one blank after the action is a valid e-mail address (See the recipient address description for the allowed formats).

Example:

```
* Sample XMITIP AddressFile Address List
To first.last@dummy.com
Cc second.last@dummy.com
Bcc third.last3@dummy.org
Cc "Fourth Last" fourth.last@dummy.com
From first.last@from.com
ReplyTo reply_first.last@reply.com
```

Note that the address list is case insensitive. The text can be all upper case, all lower case, or mixed case with no problems.

Use of qualified addresses is also allowed.

### **ASA**

ASA instructs XMITIP to assume that the file attachments contain ASA carriage control. The default is to use the carriage control defined in the DCB of the input data set. This keyword should be used if FILEDD is used and refers to a data set on VIO.

(see MACH and IGNORECC).

### **BCC**

BCC is the Blind Carbon Copy address keyword. The syntax is:

BCC [first.last@dummy.com](mailto:first.last@dummy.com)

Or

BCC ([first.last@dummy.com](mailto:first.last@dummy.com) [second.last@dummy.org](mailto:second.last@dummy.org) [third.last@dummy.com](mailto:third.last@dummy.com) ...)

Note that with the BCC the qualified name (within quotes) is not required or used.

### **CC**

CC is the keyword for Carbon Copy. The syntax is similar to that for BCC except that the qualified name may be coded as it will be used if provided:

CC [first.last@dummy.com](mailto:first.last@dummy.com)

Or

CC "First Last" <[first.last@dummy.com](mailto:first.last@dummy.com)>

Or

CC ("First Last" <[first.last@dummy.com](mailto:first.last@dummy.com)> "Second Last" <[second.last@dummy.org](mailto:second.last@dummy.org)> [third.last@dummy.com](mailto:third.last@dummy.com) ...)

### **CONFIG**

This option allows the user to define a XMITIP configuration file containing XMITIP keywords and options for the current XMITIP execution. The configuration file must be a sequential dataset or a member of a PDS.

CONFIG data-set-name

CONFIG may be abbreviated as CFG.

Notes:

- Only data in columns 1 to 72 are used (to avoid sequence numbers)
- A '+' or '-' may be used as a continuation character if desired but it is optional
- Keywords and options may span records
- Any keywords used will over-ride any specified on the XMITIP command
- Both CONFIG and CONFIGDD may be specified multiple times in one command
- Both CONFIG and CONFIGDD are recursive (you can include them in other configuration files)
- The last occurrence of a keyword is the one that is used
- An \* in column 1 indicates a comment

**CONFIGDD**

This option allows the specification of a DDname for the XMITIP configuration file for the current execution. See **CONFIG** for details.

CONFIGDD ddname

CONFIGDD may be abbreviated as CFGDD.

**DEBUG**

This option is used when you want to see the generated SMTP control statements. The message will not be sent, instead every control statement along with the requested message and attachments, will be written to the screen (for TSO use) or to the SYSTSPRT DD when used in batch.

**EMSG**

This option must be used under ISPF and with a MSGDS dsn.

**ERRORSTO**

Use this option to send an e-mail to the specified address of any delivery errors. The syntax is:

ERRORSTO [first.last@dummy.com](mailto:first.last@dummy.com)

**FILE**

The FILE keyword describes the sequential data set or member of a partitioned data set that will be included with the electronic mail as an attachment. The syntax is:

FILE data-set-name

Or

FILE (data-set-name1 data-set-name2 ...)

Note that the data set name may also be a generation data set such as hlq.data.set(-2) but it may not reside on tape.

If the data set is a partitioned dataset a member name must be specified. The member name may be a specific member name or a mask using a \*. For example:

|     |  |
|-----|--|
| AB* | will send all members starting with AB |
| *   | will send all members                  |

Abbreviation: FI

**FILEDD**

The FILEDD keyword describes the DD statements that reference the sequential data set or member of a partitioned data set that will be included with the electronic mail as an attachment. The syntax is:

FILEDD ddname

Or

FILEDD (ddname1 ddname2 ...)

Note that a DDname may be a concatenation of data sets.

**Warning:** If FILEDD refers to a temporary data set that resides in VIO the file conversion to HTML, RTF, or PDF the use of **ASA** or **MACH** should be used if the data set contains carriage control..

**FILEDESC**

The FILEDESC is a set of characters with no intervening spaces that is used to describe the file attachment. Not all electronic mail packages support this information so it is not frequently used. The syntax is:

FILEDESC file-description-for-a-single-file

Or

FILEDESC (file-description-1 file-description-2 ...)

The FILEDESC value may contain any or all of the supported symbolic variables (see list later).

Note that since blanks are not allowed you can use dashes (-) or underscores (\_) or other special characters except parentheses.

**FILENAME**

The FILENAME keyword describes the name that the data sets referenced by FILE, FILEDD, or FILEO will be known as when attached to the e-mail. The syntax is:

FILENAME file1.txt

Or

FILENAME (file1.txt file2.rtf file3.pdf ...)

FILENAME may be abbreviated as FILEN.

Note that the use of FILENAME will override any suffix specified in the FORMAT keyword for the attached file.

The FILENAME value may contain any or all of the supported symbolic variables (see list later).

If the FILENAME contains embedded blanks then it must be enclosed within parenthesis and single quotes.

e.g.       FILENAME ('sample file.txt')  
          or       FILENAME('sample file.txt' 'file 2.pdf' file3.txt)

**FILEO**

The FILEO keyword describes a data set which resides within an MVS Open Edition (aka z/OS Unix System Services) HFS. The syntax is:

FILEO '/etc/profile'

Or

FILEO ('/etc/profile' '/etc/rc.config' ...)

**FOLLOWUP**

The FOLLOWUP keyword will generate a Internet Calendar (iCalendar) file attachment that will create a ToDo entry in the users Calendar or ToDo calendaring tool (assuming the mail client supports the iCalendar files). The syntax is:

FOLLOWUP date

Where   date is in the format mmddyy  
          Or +nnn where nnn is the number of days from today

FOLLOWUP may be abbreviated as FUP.

## **FORMAT**

The **FORMAT** keyword is the most complex keyword used in XMITIP. This keyword defines the formatting that will be used for the **FILE**, **FILEDD**, and **FILEO** data sets when they are attached to the e-mail.

Note that all data on the mainframe is stored in the EBCDIC character set and is translated to the ASCII character set during the transmission. Any data that should not be translated should be attached in Binary format.

There must be one format string (type/option) per file attachment with the default being a plain text attachment. If you want to use the same format for all file attachments then the type must be preceded by an \* (see below in the syntax).

The syntax is:

FORMAT type/option

Or

FORMAT (type/option type2/option2 ...)

Or

FORMAT \*type/option indicates to use this format for **all** file attachments

### ***FORMAT TXT***

This is the default for any attachment unless another format is specified.

### ***FORMAT CSV***

The input file must be in Comma Separated Value (CSV) format and will be attached as a text file attachment with a CSV suffix.

### ***FORMAT BIN***

This format is used when attaching data that should be sent without translation.

### ***FORMAT GIF***

This format is used when attaching a graphic interchange format (GIF) binary file. The input must already be in GIF format.

### ***FORMAT ICAL***

This format indicates that an Internet Calendar file is being attached. The file suffix for the filename must be *ics*.

Note that under the ISPF dialog using a \* in the file attachment field with a format of ICAL will bring up a prompting panel to generate the iCalendar file.

### ***FORMAT HTML***

This format indicates that the file will be converted to a basic HTML format that is easily viewed using any Web Browser (e.g. Netscape, Microsoft Internet Explorer). This option supports several parameters. A null may be used in place of a parameter to take a default. If the input file is already in HTML format then no conversion is performed and the file is just attached with the appropriate information.

FORMAT HTML/color/suffix/font-size/banner/table/header/wrap/semicolon

Or

FORMAT HTML/color-color/suffix/font-size/banner/table/header/wrap/semicolon

Or

FORMAT HTML/DS:configuration-data-set

Or

FORMAT HTML/DD:configuration-ddname

Color defines the background color of the web page when only a single color is specified. When two colors are specified then the first color is the background color and the second color is the text (or foreground) color.

The colors that may be used are denoted below. The abbreviation is the smallest number of characters that uniquely identify the color:

| Color   | Abbreviation | Color  | Abbreviation |
|---------|--------------|--------|--------------|
| Aqua    | A            | Navy   | N            |
| Black   | Bla          | Olive  | O            |
| Blue    | Blu          | Purple | P            |
| Fuchsia | F            | Red    | R            |
| Gray    | Gra          | Silver | S            |
| Green   | Gre          | Teal   | T            |
| Lime    | L            | White  | W            |
| Maroon  | M            | Yellow | Y            |

**Suffix** can be used to define the file suffix that will be used for the attached file, but only if no FILENAME keyword is specified for this file.

**Font-size** defines a relative font size ranging from 1, which is extremely small, to 7, which is very very large. The font sizes are relative because the user can alter the font size defaults within their Web Browser.

**Banner** may be Yes or No (or null) and indicates if the file-description is to be used as a Banner in the generated html document. If there is no file description then the file dataset name will be used.

**Table** may be Yes or No (or null) and indicates that the input file is a CSV file that should be converted to an HTML table.

**Header** may be Yes or No (or null) and indicates that the input CSV file contains a header row.

**Wrap** may be Yes or No (or null) and indicates that the text in each cell in the HTML table should wrap.

**Semicolon** may be Yes or No (or null) and indicates that a semicolon (;) has been used in the CSV file instead of a comma (,).

Example:

FORMAT HTML/w-bla//4

This keyword will format a file as HTML with the background color white and the text color black. The font size will be 4.

The configuration file is any sequential data set (or PDS member) with TXT2HTML statements coded in each record. An \* in column 1 indicates a comment record.

- Notes:
1. Beware of sequence numbers
  2. TITLE is NOT allowed in a Configuration file

***FORMAT PDF***

PDF is short for Portable Document Format and is the file format used by the Adobe Acrobat reader. A version of Adobe Acrobat is available free for the Linux, Macintosh and Windows workstations. The syntax for this is:

FORMAT PDF/layout/font-size/paper-size/lpi/rpo

Or

FORMAT PDF/txt2pdf-configuration-file

Where:

Layout defines the paper orientation of Portrait or Landscape. These can be abbreviated as just P or L.



Font-size is the size of the font in points. There are 72 points to an inch. The default is 9.

Paper-size defines the width and height of the paper. Valid options are Letter (LET), Legal (LEG), A4, and user defined widthXheight (e.g. 4x6).

LPI is lines per inch. The default is 8 lines per inch. If the font size is adjusted this value should be reviewed.

RPO, short for Read and Print Only determines if the created PDF file will be secure. The syntax for this option is:

|     |                  |
|-----|------------------|
| No  | No security      |
| 40  | 40-bit security  |
| 128 | 128-bit security |

or

nn:upw where **nn** is 40 or 128 and **upw** is the User Password required for the user to read the file

The txt2pdf-configuration-file may be specified as DS:data-set-name or DD:ddname. If the ddname option is used then the DD will have to be pre-allocated to the TSO session or Batch JCL. Use the TXT2PDF ISPF Dialog to generate the configuration file for simplicity.

Important: If using this option then all PDF formatting options must be included in the configuration file.

Note: the MARGIN keyword can be used with this format to adjust the top, bottom, left, and right margins on the page.

Examples:

FORMAT PDF/1/9/let/6  
or  
FORMAT PDF/1/9/let/6/40:readpw

Defines landscape orientation, 9-point font, letter size paper, and 6 lines per inch on the first and on the second the same formatting with 40-bit encryption and a read only password.

### ***FORMAT RTF***

RTF is short for Rich Document Format and is a common document interchange format that can be used by Microsoft Wordpad, Microsoft Word, WordPerfect, Star Office and other word processors. The syntax for this is:

FORMAT RTF/layout/font-size/paper-size/ro  
Or  
FORMAT RTF/txt2rtf-configuration-file

Where:

Layout defines the paper orientation of Portrait or Landscape. These can be abbreviated as just P or L.

Font-size is the size of the font in points. There are 72 points to an inch. The default is 9.

Paper-size defines the width and height of the paper. Valid options are Letter (LET), Legal (LEG), A4, and user defined widthXheight (e.g. 4x6). Note: the units for the Width and Height is defined for each installation and may be either Inches or Centimeters. Check with your installation support if you are not sure.

ro is Yes to enable ReadOnly mode or No (or blank) for normal Read/Write mode.

The txt2rtf-configuration-file may be specified as DS:data-set-name or DD:ddname. If the ddname option is used then the DD will have to be pre-allocated to the TSO session or Batch JCL.

Important: If using this option then all RTF formatting options must be included in the configuration file.

Sample RTF Configuration File:

```
CC
FONT 9
MARGINS .8/.8.5.5
```

Note: the MARGIN keyword can be used with this format to adjust the top, bottom, left, and right margins on the page.

Example:

FORMAT RTF/p/9/let

Defines a document with portrait paper orientation, 9-point font, and letter size paper.

#### ***FORMAT XMIT***

This format is used when sending a dataset created by the TSO Transmit command. The dataset will be attached as a binary file attachment. If the input is not in TSO Transmit format then XMITIP will use TSO Transmit to put it into XMIT format.

#### ***FORMAT ZIP***

This format will cause the text file to be compressed using a ZIP utility (PKZIP/MVS, ISPFZip, or InfoZip) if installed and available. The syntax is:

FORMAT ZIP/name-in-archive

The name-in-archive is used by InfoZip and PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment. For InfoZip the name-in-archive will have a installation defined high-level-qualifier appended to its front.

#### ***FORMAT ZIPBIN***

This format will cause the binary (no translation will be performed) file to be compressed using a ZIP utility (PKZIP/MVS, ISPFZip, or InfoZip) if installed and available. The syntax is:

FORMAT ZIPBIN/name-in-archive

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment.

#### ***FORMAT ZIPCSV***

This format will cause the input file, which must already be in CSV format, to be compressed using a ZIP utility and then attached. The syntax is:

FORMAT ZIPCSV/name-in-archive

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment.

**FORMAT ZIPGIF**

This format will cause the input file, which must already be in GIF format, to be compressed using a ZIP utility and then attached. The syntax is:

FORMAT ZIPGIF/name-in-archive

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment

**FORMAT ZIPHTML**

This format will cause the file to be converted to HTML format and then to be compressed using a ZIP utility (PKZIP/MVS, ISPZip, or InfoZip) if installed and available. The syntax is:

FORMAT ZIPHTML/name-in-archive/color/font-size/banner/table/header/wrap/semicolon

Or

FORMAT ZIPHTML/name-in-archive/DS:configuration-data-set

Or

FORMAT ZIPHTML/name-in-archive/DD:configuration-ddname

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment.

See the FORMAT HTML description above for the definition of the color and font-size parameters.

**FORMAT ZIPPDF**

This format will cause the file to be converted to PDF format and then to be compressed using a ZIP utility (PKZIP/MVS, ISPZip, or InfoZip) if installed and available. The syntax is:

FORMAT ZIPPDF/name-in-archive/layout/font-size/paper-size/lpi/rpo

Or

FORMAT ZIPPDF/name-in-archive/txt2pdf-configuration-file

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment.

See the FORMAT PDF description above for the definition of the layout, font-size, paper-size, lpi parameters and the txt2pdf-configuration-file option.

**FORMAT ZIPRTF**

This format will cause the file to be converted to RTF format and then to be compressed using a ZIP utility (PKZIP/MVS, ISPZip, or InfoZip) if installed and available. The syntax is:

FORMAT ZIPRTF/name-in-archive/layout/font-size/paper-size/ro

Or

FORMAT ZIPRTF/name-in-archive/txt2rtf-configuration-file

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment.

See the FORMAT RTF description above for the definition of the layout, font-size, and paper-size parameters and for the information on the specification for the configuration file.

### **FORMAT ZIPXMIT**

This format will cause the input file, which must already be in TSO Transmit (XMIT) format to be compressed using a ZIP utility and then attached. If the input is not in TSO Transmit format then XMITIP will use TSO Transmit to put it into XMIT format. The syntax is:

FORMAT ZIPXMIT/name-in-archive

The name-in-archive is used by PKZIP/MVS version 2.51 and higher and is similar to the FILENAME keyword. If specified the file will be placed in the ZIP archive file using this name. The FILENAME keyword is used to specify the name the ZIP archive will be known as in the e-mail attachment

**Note:** If no FORMAT keyword is used then the format used will be determined by:

1. looking at the filename suffix (e.g. filename.rtf will use format of rtf)
2. looking at the attachment dsn suffix if there is no format or filename (e.g. dataset.rtf will use format of rtf)

Abbreviation: FORM

### **FROM**

The FROM keyword defines the electronic mail address of the sender. The syntax is:

FROM [e-mail@address](#)

The format of the e-mail@address can be seen above under the To Address.

Examples:

FROM [first.last@company.com](#)  
From: "first last" [first.last@company.com](#)

Abbreviation: FR

**IMPORTANT:** This keyword should always be specified and should be your personal e-mail address on your official electronic mail system. This address will be used if any of the To, Bcc, or CC addresses are incorrect and the e-mail bounces back. If not used the bounced e-mail will be routed to the senders TSO system – and you do not want to read e-mail on TSO since there is no good e-mail reader on TSO and TSO does not support file attachments.

### **HLQ**

This keyword defines a high level qualifier that will be used for work data sets used within the XMITIP application. This keyword should be used when running XMITIP in batch under userids that do not have a TSO segment defined. The syntax is:

HLQ xxxx  
Or  
HLQ xxx.yyy

### **HTML**

This keyword has no parameters and simply indicates that the text in the MSGDS or MSGDD is coded in HTML or Rich Text format. Use this if you want to send text that has color, bolding, italics, or different font sizes.

### **IDVAL**

This keyword when used will enable e-mail address verification providing that the XMITLDAP defaults module has been updated appropriately. With this all e-mail addresses that fail validation will cause the e-mail to be aborted.

**IMPORTANCE**

This keyword is used to define the importance of the e-mail. This tag will cause some e-mail systems to mark or flag the e-mail with an importance icon. The syntax is:

IMPORTANCE parameter

Where parameter may be one of these values: High, Normal, or Low.

**IGNORECC**

Bypass all carriage control processing. The default is to use the carriage control defined in the DCB of the input data set.

(see ASA and MACH).

**IGNOREENC**

This keyword will cause XMITIP to ignore any installation defined encoding.

**IGNORESUFFIX**

This keyword will prevent XMITIP from automatically adding a FORMAT friendly suffix to the FILENAME if the user specified suffix is not appropriate for the specified FORMAT.

**LANG**

This option defines a default language to be used to translate the date (month and day). Check with your systems programmer for the supported languages. The standard supported languages are Brazilian, Dutch, French, German, and Spanish. The default is English.

LANG Spanish

**MACH**

MACH instructs XMITIP to assume that the file attachments contain Machine carriage control. The default is to use the carriage control defined in the DCB of the input data set. This keyword should be used if FILEDD is used and refers to a data set on VIO.

(see ASA and IGNORECC).

**MARGIN**

The MARGIN keyword defines the top, bottom, left, and right margins for both the PDF and RTF formatted file attachments. The syntax is:

MARGIN left/right/top/bottom

Or

MARGIN (left/right/top/bottom left2/right2/top2/bottom2 ...)

Note: the units for the Margin specification is defined for each installation and may be either Inches or Centimeters. Check with your installation support if you are not sure.

If the form of FORMAT is used of FORMAT \*xxx/... which applies the FORMAT to all file attachments then a single MARGIN specification is used for all file attachments as well.

**MSGDD**

The MSGDD defines a DDName that references a sequential data set or member of a partitioned data set that contains the text to be included in the message portion of the e-mail. Its use is mutually exclusive of the other MSGxx keywords. The syntax is:

MSGDD ddname

Note that the DD statement may be a DD \* and/or a concatenation of multiple data sets.

## **MSGDS**

The MSGDS defines a data set name, sequential or a member of a partitioned data set, which contains text that will be included in the message portion of the e-mail. Its use is mutually exclusive of the other MSGxx keywords. The syntax is:

MSGDS data-set-name

Or

MSGDS data-set-name(member)

Or

MSGDS omvs.file-name

Or

MSGDS \* (under ISPF Only)

When MSGDS \* is used under ISPF the ISPF Editor is opened on an empty data set (record length is 72) in which the message text may be entered (or copied from another data set).

Note that when using an OMVS file name that the file name is case sensitive (e.g. file ABC is not the same as AbC).

## **MSGQ**

The MSGQ keyword may be used to pull the message text from the TSO Stack. This option is designed to be used primarily by automation tools that will QUEUE lines of text to the TSO Stack to be sent in a e-mail. Its use is mutually exclusive of the other MSGxx keywords. The syntax is:

MSGQ

## **MSGT**

The MSGT keyword may be used to include message text within the XMITIP command. The message text must be enclosed in either single quotes ' or double quotes ". A \ may be used to cause the following text to be placed on a new line. Any of the supported symbolics may be used within the message text. The use of the escape character, a /, prior to the \ will allow the \ to be included within the message.

MSGT 'this is a test of the message text keyword'

If the first 5 characters of the message text is **html:** then the string **html:** will be removed and the message text will be processed as html text.

## **MSG72**

This keyword is used in conjunction with MSGDD , MSGDS, MSGQ, and MSGT to limit the data processed to the first 72 columns. This is useful if the message data contain sequence numbers in columns 73 to 80. The syntax is:

MSG72

## **MURPHY**

This keyword will cause a cute or humorous quote to be included in the message of the e-mail. It will appear after the message text and before a signature. The syntax is:

MURPHY

## **NOCONFIRM**

This keyword is useful if you do not want to see the XMITIP report. Its use is not recommended. The syntax is:

NOCONFIRM

**NOEMPTY**

This keyword over-rides the system configuration default. NOEMPTY will force XMITIP to terminate if it encounters an empty file attachment.

NOEMPTY**NOIDVAL**

This keyword is dependent upon the LDAP e-mail address validation being enabled. If this validation is enabled then this keyword will cause XMITIP to not validate any of the e-mail addresses specified. The syntax is:

NOIDVAL**NOMSG**

The NOMSG keyword is used for documentation purposes only. When you are sending a e-mail with a file attachment with no message this keyword can be used so that others reviewing your command syntax will know that you explicitly decided not to send a message with the file. The syntax is:

NOMSG**NORTFXLATE**

This option should be used only if you have RTF fields in your document and you do not want XMITIP to insert escape characters in front of them. The syntax is:

NORTFXLATE**NOSPOOF**

The NOSPOOF keyword can be used only if it is enabled by the installation to bypass the generation of the AntiSpooof message text in the message.

**NOSTRIP**

The NOSTRIP keyword is used to turn OFF the removal of trailing blanks in file attachments. This may be necessary if the receiving user needs to process the data and requires the trailing blanks. The syntax is:

NOSTRIP

Usage Notes:

1. NoStrip may not work in all environments for Format types of HTML, RTF, or TXT.
2. NoStrip will always work for Format types of Binary, XMIT, and all formats of ZIP.
3. NoStrip does not work for Format PDF.

**PAGE**

The PAGE keyword is used to generate a short message to a pager and defines the message text to be used with the e-mail. The syntax is:

PAGE 'page message text'

The text must be enclosed in either single (') or double (") quotes. Using a double quote (") on the outside will allow the use of a single quote (') on the inside. The size of the paging text is not limited.

The PAGE may contain any or all of the supported symbolic variables (see list later).

Examples:

PAGE 'message text'

PAGE "sample message with single quote (') inside"

PAGE 'JOB &job completed on &date at &time'

**Notes**

1. That the use of PAGE precludes the use of SUBJECT, any of the MSG keywords and any of the FILE keywords.
2. The recipient address should be to a pager. Many paging providers have a e-mail address of the format [pager-phone-number@paging.provider.com](mailto:pager-phone-number@paging.provider.com) for example: 1115551212@paging.com
3. The &RC or &RCH symbolic can be useful when sending a page about the abnormal completion of a JOB (do not use &RCA as it generates too much information)

### **PDFIDX**

This keyword defines the indexing information for the Adobe Acrobat PDF file attachments. The syntax is:

PDFIDX row/column/length

Where

|        |   |
|--------|---|
| Row    | The row on the page where the index is to be found. If 0 then look on all rows. |
| Column | The column where the index is to be found.                                      |
| Length | The length of the index   |

### **PRIORITY**

This keyword is similar to IMPORTANCE but some e-mail packages flag priority differently than importance. The syntax is:

PRIORITY parameter

Where parameter is one of these values: Urgent, Normal, or Non-Urgent.

Abbreviation: PR

### **RC0**

This keyword is used when the input FILE, FILEDD, etc. might be empty and the user requires a zero return code.

### **RECEIPT**

This keyword will cause an e-mail to be returned to the specified address when the e-mail has been received. Some e-mail systems may send the receipt when the e-mail is opened, others when it is delivered to the users inbox (but not opened yet). The syntax is:

RECEIPT address

Where address is the e-mail address to send the RECEIPT.

### **REPLYTO**

The REPLYTO keyword can be used when you want any replies to the e-mail to be delivered to someone else. The syntax is:

REPLYTO address

The address is any of the valid electronic mail address formats – see the To Address above for address syntax information.

Abbreviation: REP

### **RESPOND**

This option will generate in the message text of the e-mail an automatic reply with the text of the respond option.

Syntax:           RESPOND option  
           Or           RESPOND (option1 option2 ...)



The message text will be generated with HTML tags (see the HTML option) and the response links will generate an e-mail with the subject changed to “re: subject: option” and the body of the e-mail response will contain the option text.

Note that the option must be a single word with no spaces.

### **SENSITIVITY**

This keyword is similar to IMPORTANCE and PRIORITY in that it will cause a tag to be generated that some e-mail packages will use to flag the e-mail. The syntax is:

SENSITIVITY parameter

Where parameter is one of these values: **Private**, **Personal**, **Company-Confidential** or **Confidential** each of which may be abbreviated to:

**PR** for Private  
**PE** for Personal  
**COM** for Company-Confidential  
**CON** for Confidential

If this parameter is coded a line of text may be included with each e-mail at the beginning or end of the message text based upon an installation option. The line will be one of the following:

This E-Mail is Confidential and is intended for use by the recipient(s) only.

This E-Mail is Company-Confidential.

This E-Mail is Personal and is intended for use by the recipient(s) only.

This E-Mail is Private and is intended for use by the recipient(s) only.

Abbreviation: SENS

### **SIG**

The SIG keyword references a sequential data set or member of a partitioned data set, which contains text to be used as a signature. The syntax is:

SIG data-set-name

### **SIGDD**

The SIGDD references a DD statement that refers to a sequential data set, or member of a partitioned data set, which contains text to be used as a signature. The syntax is:

SIGDD ddname

### **SUBJECT**

The SUBJECT keyword defines the subject to be used with the e-mail. The syntax is:

SUBJECT ‘subject text’

The subject must be enclosed in either single (‘) or double (“) quotes. Using a double quote (“) on the outside will allow the use of a single quote (‘) on the inside. The size of the subject is not limited.

The SUBJECT may contain any or all of the supported symbolic variables (see list later).

Examples:

SUBJECT ‘sample subject’

SUBJECT “sample subject with single quote (‘) inside”

SUBJECT ‘Report for &date-7 to &date Created on &day at &time’

Abbreviation: SUB

**TLS**

Defines if the e-mail is to be transmitted in a secure format using TLS.

Options are ON or OFF.

Note this will not over-ride any setting in the CSSMTP server.

**TPAGELEN**

Defines the maximum number of characters allowed in a text page. If 0 then there is no limit.

**VIANJE**

Defines a NJE Node where the e-mail is to be routed where a SMTP server will pick it up from the JES SPOOL to send.

VIANJE value

**ZIPMETHOD**

The ZIPMETHOD keyword defines the compression method to use if other than the default compression is desired. The syntax is:

ZIPMETHOD value

The value depends upon the ZIP utility used:

PKZIP/MVS supports Normal, Fast, Maximum, and Store.

ISPZIP does not support this option.

INFOZIP supports values of 0 through 9 (0=store only 9=max compression).

**ZIPPASS**

The ZIPPASS is currently only supported by INFOZIP and PKZIP/MVS. The password may be up to 64 characters in length (no blanks allowed because of an XMITIP limitation). The syntax is:

ZIPPASS password-for-zip

Note: This password is not totally secure. It is not approved for high security environments.

If using PKZIP Version 5.6 or newer the security of the ZIP file can be enhanced by using AES encryption. To do so the format of the password is:

xxx:passsword

where xxx: is 128, 192, or 256 and the : is a required separator.

## Supported Symbolic Variables

The following are the supported symbolic variables that are allowed in the FILENAME, FILEDESC, MSGT, and SUBJECT keywords:

|   |  |
|---|--|
| &DATE or &DATE-n                              | replaced by the current date or current date-n days<br>In Subject and MSGT replaced with <i>month dd, yyyy</i><br>Other uses replaced with <i>yymmdd</i> |
| &DAY or &DAY-n                                | replaced by the current day of the week or the current day- day of the week  |
| &CDATE  | replaced by current date <i>yyyymmdd</i>   |
| &EDATE or &EDATE-n                            | replaced by the current or calculated date in European format (dd/mm/yy)   |
| &JDATE or &JDATE-n                            | replaced by the current or calculated Julian date  |
| &JOB  | replaced by the current job name   |
| &JOBID  | replaced by the current job number (e.g. JOBnnnnn)   |
| &JOBNUM                                       | replaced by the current job number (e.g. Jnnnnn)   |
| &MONTH or &MONTH-n                            | replaced by the name of the current month or the calculated month  |
| &MM   | replaced by 2 digit month (e.g. 01 to 12)  |
| &PLR  | Left Parenthesis   |
| &PRR  | Right Parenthesis  |
| &RC   | replaced by the condition code for the last executed step.   |
| &RCA  | replaced by all of the step completion codes for the job prior to the XMITIP step (not recommended for use with PAGE or SUBJECT).                        |
| &RCH  | Replaced by the highest condition code of all the previous steps   |
| &SDATE or &SDATE-n                            | replaced by the current or calculated short date ( <i>mmdd</i> )   |
| &SYSID  | replaced by the current system name  |
| &SYSNAME                                      | replaced by the current system name  |
| &SYSPLEX                                      | replaced by the name of the current sysplex  |
| &TIME   | replaced by the current time   |
| &CTIME  | replaced by a compact time ( <i>hhmmss</i> )   |
| &TIMEZONE                                     | Current Timezone (e.g. PDT or timezone offset) based upon local customization  |
| &UDATE or &UDATE-n                            | replaced by the current or calculated date ( <i>mmdyy</i> )  |
| &USERID                                       | replaced by the current userid   |
| &IWEEK or &IWEEK-n                            | replaced by the week number ( <i>ww</i> )  |
| &IWEEKE or &IWEEKE-n                          | replaced by the week number ( <i>yyyy-Www-d</i> )  |
| &IWEEKR or &IWEEKR-n                          | replaced by the week number ( <i>yyyy-Www</i> )  |
| &XMITVER                                      | version of XMITIP being used   |
| &YEAR2 or &YEAR2-n                            | replaced by either a 2 digit year ( <i>yy</i> ) or yy-n  |
| &YEAR or &YEAR-n<br>And<br>&YEAR4 or &YEAR4-n | replaced by either a 4 digit year ( <i>yyyy</i> ) or yyyy-n  |
| &Z  | preceding blank and &Z are eliminated thus allowing concatenation  |

Minimum required syntax:

The minimum required syntax is a to-address and a message or a file specification.

```
XMITIP user@domain.com MSGDS data.set.name
```

Or

```
XMITIP user@domain.com FILE data.set.name
```

The sample JCL is an execution of the TSO Batch Terminal Monitor Program (or TMP). The dash (-) or a plus (+) is used to indicate that a continuation follows when a command cannot fit on a single card image.

Data set coding conventions are the same as for any TSO Batch TMP invocation. The data set must be fully qualified within single quotation marks otherwise the userid under which the job is submitted will be appended as the high level qualifier. Members of partitioned data sets may be distributed as well by coding the member name within parentheses as is the common JCL coding standard.

Note that the command is completely case insensitive.

## XMITIP Examples

These examples illustrate some of the capabilities of the XMITIP application. The best way to learn about XMITIP is to use the ISPF Dialog to experiment. The dialog generates a report of the XMITIP command that it generated as well as providing a BATCH option which generates a data set that you can easily copy for use in a batch job or use in a TSO or ISPF application.

**Note:** when using XMITIP to e-mail using a distribution list that you should use the AddressFile or AddressFileDD keywords and data sets. This avoids having a lot of addresses in the command itself and eliminates the potential of causing a syntax error when changing addresses as recipients are added or removed.

These examples demonstrate using the Batch TMP (Terminal Monitor Program) to invoke TSO commands in batch. Some things to be aware of are:

1. The data set referenced by the //SYSEXEC statement is the location where the XMITIP application has been installed.
2. When the command exceeds one statement it must be continued with either a + or a – on the statement immediately before the continuation statement.
3. The case of the command is not relevant. It can be all upper case, all lower case, or mixed case. Note that the subject is the only information in which you may be concerned about case.
4. Not all examples include a FROM keyword, however it is always good practice to include this keyword with your primary e-mail address coded.
5. Use of a – for continuation will yield extra spacing which may not be desirable in a long subject while the use of the + for continuation will suppress the extra spacing.

### Example 1: Send a PDS Member with No Message

This example will send a member of a partitioned data set with no message text and a short subject to 2 users.

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP (user1@domain.com user2@domain.com) nomsg file -
    'hlq.data.set(member)' -
    from sender@domain.com -
    subject 'send a file but no message'
```

### Example 2: Send a message to One Address.

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP user@domain.com msgds my.message.text -
    from sender@domain.com -
    'urgent message'
```

### Example 3: Send a Message with CC and FROM specified.

Sending example 2 with 2 CCs with a From of my primary e-mail so questions about what I'm sending can be sent there.

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP user@domain.com msgds my.message.text subject -
    'urgent message' -
    cc (user2@domain.com user3@domain.com) -
    from mynotes@domain.com
/*
```

### Example 4: Send a Single PDS Member To One Address.

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP user@domain.com nomsg file 'hlq.data.set(member)' -
    from sender@domain.com -
    subject 'pds member'
/*
```

### Example 5: Send a Dataset in Landscape with a 9 point font.

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//REPORT DD DISP=(OLD,DELETE),DSN=&&RPT
//SYSTSIN DD *
%XMITIP user@domain.com nomsg filedd report subject -
    'passed report' -
    from sender@domain.com -
    format rtf/land/9
/*
```

### Example 6: Send a Passed Dataset to a List using Blind Copies.

```
//REPORT EXEC PGM=DOREPORT
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=data.base
//OUTPUT DD DISP=(,PASS),UNIT=SYSDA,DSN=&&RPT
/*
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//REPORT DD DISP=(OLD,DELETE),DSN=&&RPT
//SYSTSIN DD *
%XMITIP *report-list* bcc user@domain.com nomsg filedd report -
    subject 'report distribution' -
    from myname@domain.com
/*
```

### Example 7: Send a Dataset Using an AddressFile Dataset:

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP * nomsg file 'hlq.data.set(member)' -
        subject 'pds member' -
        from myname@domain.com -
        Addressfile 'hlq.data.set(address)'
/*
```

Where 'hlq.data.set(address)' contains the following:

```
To user1@domain.com
CC user2@domain.org
Cc user3@domain.ca
Bcc user4@domain.uk
```

### Example 8: Using MSGQ

This example demonstrates a REXX Exec that calls XMITIP with the MSGQ option.

```
/* rexx */

queue "this is a test 1"
queue "this is a test 2"
queue "this is a test 3"
"%xmitip user@domain.com subject 'test of msgq' " ,
  "msgq from sender@domain.com"
```

### Example 9: Send a SYSOUT file as a Text Attachment:

What this example does is:

- create a SYSOUT file in DD SYSUT2 using IEBGENER
- Execute batch SDSF
- set PREFIX to the jobname of SDSFTEST
- set OWNER to all
- invoke Status command
- do a Find for the DD we want (SYSUT2)
- Print to File (DD) LOGFILE
- Then run the batch TMP to execute XMITIP

```
//SDSFTEST JOB acct,'Batch Job',CLASS=L,NOTIFY=&SYSUID,
//      MSGLEVEL=(1,1),MSGCLASS=X
//GENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=SYS1.PARMLIB(IEASYS00)
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//*****
//SDSFBTCH EXEC PGM=SDSF
//ISFOUT DD SYSOUT=*
//LOGFILE DD DISP=(,CATLG),UNIT=3390,DSN=userid.LOGFILE.LIST,
// SPACE=(TRK,(30,30),RLSE)
//ISFIN DD *
PRE SDSFTEST
OWNER
ST
FIND SDSFTEST
++?
FIND SYSUT2
++S
PRINT FILE LOGFILE
PRINT
PRINT CLOSE
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexex.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%XMITIP user.name@domain.com NOMSG FILE LOGFILE.LIST -
  from myname@domain.com -
  SUBJECT 'TEST WITH IEASYS00'
DELETE LOGFILE.LIST
/*
```

### Example 10: Send a Dataset in RTF Format using ZIP to save space

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexex.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%xmitip user.name@domain.com nomsg file 'hlq.weekly.report' -
  from myname@domain.com -
  subject 'weekly report' format ziprtf/weekly.rtf/port/9/let
/*
```



### Example 11: Send a Comma Separated Value (CSV) Dataset.

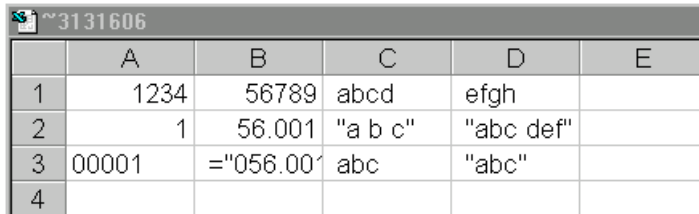
```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%xmitip user.name@domain.com nomsg file 'hlq.weekly.csv' -
    from myname@domain.com -
    subject 'CSV Test File' format txt -
    filename report.csv
/*
```

In this example we are sending a data set called 'hlq.weekly.csv', which contains comma separated value data. We are using the FILENAME keyword to explicitly name the file attachment as **report.csv** so that it will arrive with a .CSV suffix.

An example of a CSV file is:

```
01234, 56789, abcd, efgh
000001, 56.001, "a b c", "abc def"
="00001", ="056.001", abc, "abc"
```

Here is a snapshot of what this looks like in Excel.



|   | A     | B           | C       | D         | E |
|---|-------|-------------|---------|-----------|---|
| 1 | 1234  | 56789       | abcd    | efgh      |   |
| 2 | 1     | 56.001      | "a b c" | "abc def" |   |
| 3 | 00001 | = "056.001" | abc     | "abc"     |   |
| 4 |       |             |         |           |   |

Note that column A, row 3, and has leading zeros while the others cells do not. This is because of the format of the input for that cell.

### Example 12: Sample using MSGT

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
xmitip your.email@address -
    subject 'IVP2B test message text and symbolics' -
    from your.email@address -
    msgt 'This is a test +
        of the message text keyword +
        \and testing the symbolics +
        \date = &date and date-2 = &date-2 +
        \sdate = &sdate and sdate-2 = &sdate-2 +
        \update = &update and update-2 = &update-2 +
        \day = &day and day-2 = &day-2 +
        \sysid = &sysid \userid = &userid '
/*
```

Notice how the continuation lines are coded. The continuation character is the - symbol or the + symbol. The quotes around the message text are only found before the first character of the text and after the last character - not about each line of text. The \ starts a new line. Leading blanks will be included, for

example after the "This is a test" and before "of the message text". To avoid the extra blanks do not indent the message text.

### Example 13: Using a XMITIP Configuration File

```
//TSOB EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
xmitip your.email@address -
      config test.config.file -
/*
```

In this example the e-mail address is specified (as it is required as the first option of the command) and then the configuration file is specified.

### Example 14: Sending a Message to a Pager if Job Abnormally Ends

```
. . .
//TEST1 IF (RC > 4 | ABEND = TRUE) THEN
//MAIL EXEC PGM=IKJEFT1B
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
xmitip 1115551212@messaging.paging.com +
      page "job &job(&jobnum) has ended abnormally +
           on &date at &time"
/*
//      ENDIF
```

In this example if the highest return code (RC) for the previous steps is greater than 4 or the job abends then the message will be generated using the PAGE option and an e-mail address of a text pager.

### Example 15: Splitting a Report into Individual E-Mails based on a Key/Value

```
/* Invoke IEHLIST on the SYSRES Volume *
/* ----- *
//HLIST EXEC PGM=IEHLIST
//SYSPRINT DD DISP=(,PASS),UNIT=3390,SPACE=(TRK,(30,30)),
// DSN=&&LIST
//RESVOL DD DISP=SHR,UNIT=3390,VOL=SER=sysres
//SYSIN DD *
LISTVTOC VOL=3390=sysres
/* ----- *
/* BATCH TERMINAL MONITOR PROGRAM (TMP) TO EXECUTE TSO
/* APPLICATIONS IN BATCH.
/* ----- *
//TSOB EXEC PGM=IKJEFT1B,DYNAMNBR=50
//SYSEXEC DD DISP=SHR,DSN=rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//REPORT DD SYSOUT=*
//IEHLIST DD DISP=(OLD,DELETE),DSN=&&LIST
//SYSTSIN DD *
%xmitipsp dd:iehlist dd:control
/*
//CONTROL DD *
seploc=98/9/1
gmail=global.email@address
keyv=PAGE 1
keymail=your.email@address
keyv=PAGE 3
keymail=your.email@address
keyv=PAGE 10
keymail=your.email@address g
subject=Test splitter report &sepnum
msgstart
This is a test message for the XMITIP report
splitter utility.
msgend
xmitipconfig=dd:xmcc
filename=iehlist&sepnum.rtf
//XMCC DD *
FROM your.email@address +
FORMAT RTF/land/9/let +
MURPHY
/*
```

In this example only Pages 1, 3 and 10 will be e-mailed out and all other pages will be ignored.

See the documentation below for details on the control statements that are supported by this utility and for particulars on how this utility works.

The XMCC DD in the example is an example of an XMITIP configuration file.

## Example 16: PDF creation example using a PDF Configuration File

This example will demonstrate how to create a PDF from a report using the PDF Configuration DD option.

```
//???????X JOB (????????,????,1439,9999,9999),XMITIP-SAMPLE,
//          CLASS=?,MSGCLASS=?,
//          NOTIFY=???????,COND=(0,NE)
//*
//*   First step is IEBGENER creating a sample
//*   report by copying SYSUT1 data to SYSUT2.
//*
//*   Second step picks up PASSED SYSUT2 and
//*   E-Mails the report on bluebar paper.
//*
//*   Comments are at the bottom.
//*
//TESTRPT EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSIN      DD  DUMMY
//SYSUT1    DD  *
1Sample report heading
   Sample single space
   Sample to be messed up with overprint
+Sample overprint
0Sample double space
-Sample triple space
//SYSUT2    DD  UNIT=V3390,SPACE=(TRK,(15,15)),DISP=(,PASS)
//*
//EMAIL     EXEC PGM=IKJEFT1B
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN   DD  *
   %XMITIP ?????????@?????.edu +
   ASA FORMAT PDF/DD:PDFCONFIG +
   FROM UserID@OS390.?????.edu +
   SUBJECT 'Sample Report Attached &cdat' +
   MSGT 'Sample Report Attached\for &cdat.' +
   FILEDD REPORT +
   FILEDESC Sample_Report +
   FILENAME &cdat_Sample.pdf
//SYSPROC  DD  DISP=SHR,DSN=SYS?.XMITIP.EXECFB.V544
//REPORT   DD  DISP=SHR,DSN=*.TESTRPT.SYSUT2
//PDFCONFIG DD  *
ORIENT Landscape
PAPER Letter/BlueBar
BG Textmark/Bottomup/Black/Yellow/30/Sample
LM 0.5
RM 0.2
TM 0.2
BM 0.2
FONT 9/Courier/100
LPI 8
CC Yes
HLQ ???????
//*
```

## XMITIPTD – Timed Delivery Routine

The Timed Delivery routine is a REXX program that is intended to be run on a daily basis. During the once daily execution a control file is processed and those control records that match the current date will cause the defined e-mails to be sent.

XMITIPTD Syntax: %xmitiptd control-dataset debug

The control data set is a partitioned dataset (PDS) and only those members which start with TD are processed. The format of the TD members is:

- a) Each record must occur in the order defined below.

| Statement | Contents  |
|-----------|---|
| 1         | Day(s) of the week or the Day of the Month<br>e.g.<br>Sunday, Monday, Tuesday, ....<br>Or<br>5, 10, 15, 20, 30  |
| 2         | Name of the owner of this request. This is for contact purposes should questions arise about the generated e-mails or recipients                                      |
| 3         | Subject for the e-mail using the syntax:<br>SUBJECT: subject text<br><br>All XMITIP supported symbolic are allowed  |
| 4         | An XMITIP Addressfile data set name or data set name with member containing all TO, CC, and BCC addresses<br><br>Consider using a member of the control PDS for this. |
| 5         | A data set or data set with member name that contains the message text to be sent<br><br>Consider using a member of the control PDS for this.                         |
| 6         | An e-mail address that will be used as the Reply-To address to which replies and bounces will be directed.  |
| 7+        | All records after 6 are ignored.  |

NOTE: This routine is an example intended to give each installation ideas on how to do this. Should anyone develop this routine beyond this simplistic base please advise.

## XMITIPSP – File Split/Separation Utility

XMITIPSP is the XMITIP File Split and Separation Utility. It is designed to be executed in Batch using the Batch Terminal Monitor Program (TMP). The functions that are available with this utility are:

1. Split an input file into individual reports to be e-mailed based on a Key Value that is found in a specific Column and Record location.
2. Split an input file into individual reports to be e-mailed based on a Key Value that is found in a specific Column location in any record.
3. Separate an input file into individual reports to be e-mailed based on a specific number of lines per report.
4. Separate an input file into individual reports to be e-mailed based on a specific number of pages per report.
5. If separating by Key Values a COMBINE option to consolidate all separated reports for each e-mail address into a single e-mail.

XMITIPSP will perform the input file split/separation and then internally invoke XMITIP to e-mail the report subset.

An example of using XMITIPSP in Batch can be found in Example 15 above on page 35.

**Note:** when splitting by Key Value (SEPLOC and KEYV) any key values found in the input data set that are not found in a coded KEYV statement will be ignored.

### XMITIPSP Syntax

%XMITIPSP input-file control-file options

Where: Input-file is the input data set that contains the report to process.

Control-file is a data set that contains the XMITIPSP control statements

Options may be DEBUG only at this time.

If specified then the call to XMITIP will only be echoed on the report.

Both Input-File and Control-File can be coded as data set names or a DD:ddname to provide flexibility.

### XMITIPSP Control Statements

Control statements start in column 1 and all data starting from column 1 to the end of the record will be used.

Comments are indicated by an \* in column 1.

Caution: Beware of sequence numbers.

#### **COMBINE**

Combines all separated reports into single file and single e-mail based on KEYMAIL address.

The net result is that each KEYMAIL address will receive individual e-mails with a combined report of all KEYV values that match for them.

Syntax:           Combine

#### **FILENAME**

Filename defines the name the file will have when it arrives in the users' inbox. The filename may contain any of the supported XMITIP symbolics (see page 27) as well as the following two symbolics that are unique to XMITIPSP:

|         |  |
|---------|--|
| &sepval | Replaced by the current KEYV value           |
| &sepnum | Replaced by the current report subset number |

Syntax:           Filename=report.rtf

Note that the filename suffix should match the file type of any formatting that occurs. For example RTF is the suffix when using the FORMAT RTF to convert the file to rich text format.

### **GMAIL**

Is the same as KEYMAIL but is used with SEPLOC to eliminate the need to code redundant KEYMAIL addresses for multiple KEYV statements.

Each GMAIL address will be used with every KEYV statement.

Individual KEYMAIL statements may still be used with a KEYV if desired to send the separated reports to just that address for that KEYV.

Syntax:           Gmail=first.m.last@host.com

### **KEYMAIL**

Keymail defines an e-mail address that will be paired with the KEYV statement which must precede the KEYMAIL statements. There must be at least 1 KEYMAIL for every KEYV statement and may be as many KEYMAIL statements as are required.

Syntax:           [Keymail=first.m.last@host.com](mailto:Keymail=first.m.last@host.com)

See GMAIL if the same address is in every KEYV.

### **KEYPREF**

The KEYPREF is a prefix that must be found in the key value location (SEPLOC) to validate that the field contains a key.

Syntax:           Keypref=prefix

Note that the prefix is left justified with no trailing blanks and is case sensitive.

### **KEYV**

The KEYV keyword defines a Key Value that will be tested for based on the information provided in the SEPLOC keyword. Only one KEYV is allowed per Key Value and all KEYMAIL statements that are associated with this KEYV must follow immediately in the control data set.

Syntax:           Keyv=User ABC

This example demonstrates that embedded blanks may be used in a Key Value.

Masking is supported in the KEYV value:

Masking is:           ? masks any single character  
                          \* masks current and all subsequent characters

Note: May not use ? and \* in the same mask.

### **KEYSUBJ**

The KEYSUBJ defines a subject that will be used for e-mails associated with the related KEYV value.

## **MERGEMAIL**

MERGEMAIL is similar to COMBINE with the difference that with MERGEMAIL each separated report is sent as a unique file attachment instead of being combined.

## **MSGDD**

MSGDD refers to a DD statement that references a data set with message text that will be included in the generated e-mail.

Syntax:           MSGDD=ddname

## **MSGDS**

MSGDS refers to a data set that contains the text of a message to be included in the generated e-mail.

Syntax:           MSGDS=data.set.name

## **MSGSTART**

MSGSTART indicates that the statements which immediately follow are to be used as the message text for the generated e-mail:

Syntax:           MSGSTART

## **MSGEND**

MSGEND indicates that the inline message text is complete.

## **SEPLINES**

SEPLINES is used to indicate how many lines of data are to be separated into each report subset to be e-mailed.

Syntax:           SEPLINES=1000

## **SEPLOC**

SEPLOC defines the Key Value location.

Syntax:           SEPLOC=column/length/row

Where: column is the column in the data where the Key Value will be found  
length is the length of the Key Value  
row is the physical row (record) on the page where the Key Value will be found

Notes:

- Column and Length are required
- Row is optional and if not specified the Key Value will be tested on every record

## **SEPPAGES**

SEPPAGES is used to indicate how many pages of data are to be separated into each report subset to be e-mailed.

Syntax:           SEPPAGES=1000

Note: the data set must have carriage control so that page counting can occur.



## SUBJECT

The subject will be used with the generated e-mail to inform the user what they are receiving.

The subject may contain any of the supported XMITIP symbolics (see page 27) as well as the following two symbolics that are unique to XMITIPSP:

|         |  |
|---------|--|
| &sepval | Replaced by the current KEYValue<br>Note that blanks, left parens and right parens are translated to _ |
| &sepnum | Replaced by the current report subset number   |

Syntax:            Subject=this is a subject

## TO

To is used with SEPLINES or SEPPAGES to define an e-mail address to receive the split reports.

Syntax:            [TO=first.m.last@host.com](mailto:TO=first.m.last@host.com)

Multiple TO statements are allowed to provide multiple e-mail addresses to receive the split reports.

## XMITIPCONFIG

XMITIPCONFIG refers to a data set or DD which contains XMITIP configuration statements.

Syntax:            XMITIPCONFIG=data.set.name  
                    or            XMITIPCONFIG=DD:ddname

See page 12 for more information on the XMITIP configuration data set.

## XMITIPSP Required Statements

The following are the required statements for XMITIPSP execution:

### FILENAME

**KEYV** with 1 or more **KEYMAIL** statements

**SEoloc**, **SEPLINES**, or **SEPPAGES**

**TO** (if **SEPLINES** or **SEPPAGES**)

### XMITIPCONFIG

## XMITIPSP Mutually Exclusive Statements

The following statements are mutually exclusive:

**MSGDD**, **MSGDS** and **MSGSTART/MSGEND**

**SEPLINES**, **SEoloc** and **SEPPAGES**

**SEoloc** and **TO**

**COMBINE** and **SEPLINES** or **SEPPAGES**

## XMITIPSP Required JCL

XMITIPSP must be run under the Terminal Monitor Program (TMP) either in the foreground or in batch. All the standard TMP DD's are required (SYSRINT, SYSTSPRT, and SYSTSIN) along with the REPORT DD which is where the XMITIPSP processing report will be written.

## ISPF Dialog

The ISPF dialog is a very easy to use and greatly simplifies the coding of the XMITIP command, as the dialog actually reports on the complete syntax of the generated XMITIP command. The BATCH Execution Mode option can be used to generate a data set with the XMITIP generated command as well as the necessary JCL which can be submitted for a batch execution or to be copied into a production job's JCL.

Entry to the ISPF Dialog is via one of the following paths:

1. The XMITIP command, when entered with no parameters and under ISPF, will invoke the ISPF dialog.
2. The XMITIPI command will directly invoke the ISPF dialog.
3. The XMITIPFE command (see below for more information) will invoke the ISPF Dialog.
4. The XMITIPED and MAILFILE commands (see below for more information) will invoke the ISPF Dialog.

The primary ISPF panel for XMITIP is:

```
----- E-Mail Dialog 5.60 -----
Command ==>

To Address   ==>
CC Address   ==>
BCC Address  ==>
AddressFile  ==>
Subject      ==>

Message, DSN, *, ? ==>
Edit Message DSN   ==>                Yes or No

Attachment DSN or ? ==>
File Name in e-Mail ==>
Format (?=prompt)   ==>
      Settings      ==>                Yes or No

Configuration File ==>
Default Settings   ==>                Yes or No to set From, ReplyTo, etc.
Delivery Settings  ==>                Yes or No (FollowUp, Import, Prior, Sens,
                                         and Respond)

Execution Mode     ==> ISPF            I ISPF B Batch C Config P Prompt D Debug

Field level help available via PF1
```

The Execution Mode, which defaults to **ISPF**, defines whether the generated XMITIP command will be executed in the foreground under ISPF or if a batch job containing the XMITIP command will be generated. See below for more information on this.

An execution mode of **Config** will cause the generated XMITIP keywords and options to be written to the file specified in the Configuration File field. If the Configuration file already exists then a popup will prompt to replace or overlay it or to cancel and return to the XMITIP Panel to specify a different dataset name.

If the **Subject** is **PAGE** then the message text entered in the **Message DSN** field will be used for the paging text. If there is not enough room in the **Message DSN** field then enter a ? in the **Message DSN** field and a panel will popup allowing the entry of 2 lines of text.

If option **Debug** is selected then all processing will be performed with the exception that the e-mail will not be sent, the user is placed into ISPF Browse on the generated e-mail. This option can be used to review all of the SMTP and MIME control statements as well as the generated HTML and/or RTF tags if generated.

By entering FEEDBACK, or an abbreviation, in the command field the user will be placed into ISPF Edit on a form for supplying feedback (problem, question, comment, or suggestion) which will be automatically e-mailed when the Edit session ends.

Each field on the panel has field level help available by moving the cursor to that field and pressing F1.

The Default Settings option will bring up the following panel to allow you to set your personal defaults: Note that you can also select the Mail Settings option the following panel will be displayed to allow you to set the Mail delivery information:

```
----- E-Mail Settings -----
Command ==>

From Address      ==> your.from@address
Reply-To Address  ==>
Receipt Address   ==>
Signature DSN     ==> signature.dsn
Murphy            ==> Yes           Yes or No
Quiet Mode        ==> No           Yes or No for ISPF Dialog

Review/Verify then use PF3 to continue
```

If the installation has configured XMITIP to require a **From** address then this panel will display each time that the XMITIP dialog is started until the From address is filled in. **From** that point on this will only display upon request. After updating the information use PF3 to return to the primary XMITIP panel.

If you are sending a dataset as an attachment, by specifying it on the XMITIP primary ISPF panel, then the following panel will display if the attachment name changes from the prior use, or if the Settings option is requested:

```
----- E-Mail Attachment Details -----
Command ==>

For Dataset:

Attachment Desc  ==>
ZIP Settings     ==>           Yes, No, or Never
No Carriage Control ==> No           Yes or No (default=No)
Do NOT Strip Trailing Blanks ==> No           Yes or No (default=No)

Review/Verify then use PF3 to continue
```

After filling out this information use PF3 to return to the XMITIP panel.

The other panels that are displayed in this dialog are discussed below and will only appear if prompting is requested using the ? in one of the following fields:

- Recipient Address  
a ? will bring up the Address List dialog from which you can select/enter To, CC, and BCC recipients
- CC or BCC  
a ? will bring up the Address List dialog
- Attachment DSN  
a ? will bring up the File Attachment dialog
- Format  
a ? will bring up the Format prompting dialog and will then return to the XMITIP Panel with the Format field filled in with the selected format. This behavior is unique to this prompt.

## Batch JCL and XMITIP Generation from the ISPF Dialog

If the Execution Mode is Batch then the following panel will be displayed:

```
----- E-Mail JOB Submit -----
Select Processing Option: ==>

  B - Browse the generated job
  C - Copy to a data set
  E - Edit the generated job
  J - Change the JOB Statements
  S - Submit the generated job

Dataset containing the generated job is:
DSN='userid.xmitip.jcl'
```

This allows the user to review the generated XMITIP command with the following options:

|   |   |
|---|---|
| B | Browse the generated JCL and XMITIP control statements.   |
| C | Copy the generated JCL and XMITIP control statements to another data set. You could use this to copy this into a JCL library where you can further tailor it and then insert into a regularly scheduled production job. |
| E | Edit the generated JCL and XMITIP control statements. This allows you to change the JCL statements and the XMITIP control statements prior to submitting the JCL for batch execution.                                   |
| J | Use this option to modify the predefined JOB statements that are inserted into the generated JCL.   |
| S | Submit the generated JOB for batch execution.   |

This is helpful to use the ISPF dialog to build the JCL and XMITIP statements for use in a regularly scheduled batch job.

Below is an example of the generated JCL and control statements. The first four statements are the JOB statements. These statements may be modified while using the J option, or using the E (Edit) option. The data set name on the SYSEXEC data definition statement is dynamically determined based upon the location of where the XMITIPI REXX is currently being executed from.

```

Menu Utilities Compilers Help

BROWSE      SYSLBD.SYSLBD.XMITIP.JCL                Line 00000000 Col 001 080
Command ==>                                     Scroll ==> CSR
***** Top of Data *****
//SYSLBDN JOB acct,'LIONEL DYCK',CLASS=L,NOTIFY=&SYSUID,
//          MSGLEVEL=(1,1),MSGCLASS=X
//*
//*
//*-----*
//* XMITIP E-Mail JCL generated: 25 Jul 2001      *
//* Statements 1-4 are reserved for the JOB Card *
//* Verify all dsnames in the command if not     *
//* running under the generating userid.         *
//*-----*
//XMITIP EXEC PGM=IKJEFT1B,DYNAMNBR=50
//SYSEXEC DD DISP=SHR,DSN=userid.rexx.exec
//SYSPRINT DD SYSOUT=*
//SYSTSPT DD SYSOUT=*
//SYSTSIN DD *
%xmitip someone@host.com +
      Msgds jcl.cntl(msg) +
      From myemail@domain.com +
      Subject 'Batch Test' +
      Sig XMITIP.SIG +
      Murphy +
      File xmitip.pds(changes) +
      Filename changes.txt +
      Format txt
/*

```

If the execution mode of **Prompt** is requested then the following panel is displayed allowing the user to confirm sending the e-mail or cancel sending.

```

----- XMITIP Send Confirmation ----- Row 1 to 8 of 8
Command ==>                                     Scroll ==> CSR

Enter Yes to confirm Sending ==>                then press the Enter key.
or press PF3 to cancel sending of this e-mail.

Generated Command:
%xmitip lbdyck@gmail.com +
      Msgds * +
      From lbdyck@gmail.com +
      Subject 'Test e-mail' +
      Sig XMITIP.SIG +
      Murphy

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

```

If **Yes** is entered followed by the Enter key then the e-mail will be sent. If **PF3** is pressed then the e-mail will be cancelled.

Note that the entry of message text for a **MSGDS \*** will occur only after this e-mail has been confirmed to be sent.

## Address List Dialog

```
----- XMITIP E-Mail Address List ----- Row 1 of 1
Command ==>                               Scroll ==> CSR

Commands: Find Insert Cancel Sort SaveAF LoadAF LookUp
or S Select B Blind Copy C Carbon Copy D Delete I Insert R Revise U Unselect
AddressFile: Load B Browse D Delete E Edit R Revise S Select

Sel Action Recipients Name
          E-Mail Address
-
          lionel b dyck
          lbdyck@gmail.com
```

### Notes:

1. The LookUp option may not be enabled for everyone. This option, when available, will allow the user to perform a e-mail address lookup using the z/OS LDAP client referencing a LDAP server somewhere in the enterprise.
2. The SaveAF and LoadAF are used to Save an AddressFile or to Load an AddressFile.
3. The Sort commands are Sort Mail (default) or Sort Name.
4. Help is available via the F1 function key.
5. Every address specified in the Recipient To, CC, and BCC fields will be automatically added to this list for future use.

Addresses may be entered by entering I (for Insert) on the command line or in a Sel entry field. The following panel will appear to allow you to enter (or change) an Address:

```
----- XMITIP E-Mail Address Revision -----
Command: ==>

Enter or Revise the Name and Address

Recipients Name:
E-Mail Address:
```

Note: if the LookUp option is enabled then any address that is entered, which is a local address, will be validated.

Addresses are selected by an S, which places the address as a To address, by a C, which places the address in the CC address, or by B, which places the address in the BCC addresses.

If an address is selected in error a U will unselect it.

To build an addressfile:

1. Select the desired addresses using S, C, or B
2. Enter SAVEAF on the command line

This will create, or update, an AddressFile and at the same time pass this data set name to the XMITIP command that is being generated.

The LOADAF command will prompt for an AddressFile data set name and will load it into the Address List. The L selection command will load the selected AddressFile. New addresses will be inserted into the list and all addresses in the list which match will be flagged appropriately. This allows you to change the contents of an AddressFile easily, after which you should use the SAVEAF command to make the changes permanent.



A non-blank will select the desired level of formatting. When appropriate a second level of prompting will occur when additional information is needed. An example is the RTF prompt:

```

----- Format RTF Prompt -----
Command ==>
Enter/Verify

Layout          ==>          Portrait or Landscape
Font size       ==>          default is 9 (72 = 1 inch)
Paper          ==>          Letter, Legal, A4 or nnxyy
Read Only      ==> No       Yes or No for Read Only
Margins - in units of Inches
  Left ==>          Right ==>
  Top  ==>          Bottom ==>
Or TXT2RTF Configuration File:
==>

```

The PDF prompt is shown below:

```

----- Format PDF Prompt -----
Command ==>
Enter/Verify

Layout          ==>          Portrait or Landscape
Font size       ==>          default is 9 (72 = 1 inch)
                                   use sizeB for bold (9B)
Paper          ==>          Letter, Legal, A4 or nnxyy
LPI            ==>          Lines per inch (default 8)
Margins - in units of-measure
  Left ==> .3      Right ==> .3
  Top  ==> .3      Bottom ==> .3
PDF Index      ==>          row/column/length
RPO            ==>          Security (40, 128, or no)
                                   or 40:userpw 128:userpw
Or TXT2PDF Configuration File:    Create using TXT2PDF Dialog
==>

```

And the HTML Prompt:

```

----- Format HTML Prompt -----
Command ==>
Enter/Verify

Background Color ==>          see color
Text Color       ==>          chart below
Font Size        ==>          1 to 7 or null
Banner           ==>          Yes or No

Color   Abb   Color   Abb   Color   Abb   Color   Abb
Aqua    A     Gray    Gra   Navy    N     Silver  S
Black   Bla   Green   Gre   Olive   O     Teal    T
Blue    Blu   Lime    L     Purple  P     White   W
Fuchsia F     Maroon  M     Red     R     Yellow  Y

```



## XMITIP Tools

Several tools have been developed that interface to XMITIP.

### ALLOCGDG

This is a TSO command written in REXX that can (1) allocate to a specific ddname a relative GDG data set or (2) return the fully qualified name of a relative GDG.

Syntax:

```
call allocgdg ddname ""dsname(-n)""  
or  
fullname = allocgdg("*" "dsname(-n)")
```

**NOTE:**This tool does **not** work if the GDG is on Tape.

### EDIMAIL

This is an ISPF Edit macro that may be used to e-mail the file currently being edited.

### MAILFILE

This tool is designed to be used from the ISPF 3.4 data set list or from within another ISPF dialog and will invoke the XMITIP ISPF interface and prime the file attachment data set name field with the specified data set.

Syntax:

```
MAILFILE dsname
```

### MAILHFSE

This is an ISPF Edit macro that can be used to e-mail the HFS file currently being edited.

## SETSDSFK and XMITSDSF

These tools make it easy to send a SYSOUT data set as e-mail from SDSF. The SETSDSFK (Set SDSF Key) command is used to reset the value of the SDSF Function Key 6 definition so that it will invoke the XMITSDSF command. Then to use this tool:

1. Get into SDSF from within ISPF
2. Select a JOB or an individual data set (ddname)
3. Press Function Key 6
4. Follow the prompts on the XMITIP ISPF panel to send the report as an e-mail attachment.

The SETSDSFK needs to only be executed once as the function key setup is remembered in the ISPF Profile for SDSF.

Syntax:

%SETSDSFK option

Option may be:

C for CSV  
H for HTML  
P for PDF  
R for RTF  
T for Text  
Z for Zip

Note that the data set created by the action of the function key command is only temporary and the XMITSDSF will delete that data set upon completion of the e-mail.

XMITSDSF Syntax:

%XMITSDSF data-set-name option

Option may be:

H for HTML  
P for PDF  
R for RTF  
T for Text  
Z for Zip

## TXT2HTML

TXT2HTML is a Text to HTML conversion tool.

Syntax: %TXT2HTML options

Valid Options:

**IN** input-dataset or **IN DD:**input-dd  
**OUT** output-dataset or **OUT DD:**output-dd  
**COLOR** color (see XMITIP HTML color for syntax)  
**FONT** font-size (from 1 to 7)  
**CC** Yes or No (indicates if the input dataset has carriage control)  
**NOCONFIRM** (turns off all non-critical informative messages)  
**BROWSE** (browse the report if under ISPF)  
**BANNER** Yes or No (to print title as a banner)  
**TITLE** title text (must be the **last** keyword used)  
**TABLE** (to convert csv to table)  
**NOHEADER** (if no header row desired in the Table)  
**WRAP** (columns are wrapped to length of the largest word)  
**SEMICOLON** (use semicolon instead of comma)

If called with no parameters then the ISPF interface is called.

## TXT2RTF

This exec is used to convert a text file into a Rich Text Format (RTF) file that can be processed by nearly every word processor (e.g. Microsoft Word, Microsoft Wordpad, OpenOffice OfficeWriter, AbiWord).

Syntax: %TXT2RTF input output options

Valid options are:

|             |   |
|-------------|---|
| CC          | Honor Carriage Control                                |
| CONFIG      | References a configuration data set or DD:ddname      |
| FONT        | Font size (72 = 1 inch)                               |
| MARGINS     | Margins left/right/top/bottom                         |
| METRIC      | Use Centimeters for Margins instead of Inches         |
| NOCC        | Ignore Carriage Control                               |
| NOCONFIRM   | No status messages (run quietly)                      |
| NOXLATE     | Do not add escape characters for { } \ characters     |
| ORIENTATION | Paper orientation – Portrait or Landscape             |
| PAPER       | Paper size: LETTER, LEGAL, A4, or custom widthXheight |
| READONLY    | Set the RTF Read-Only flag                            |

## TXT2PDF

This exec is used to convert a text file into an Adobe Acrobat PDF file. See the TXT2PDF documentation for the syntax of this command.

## XMITIPED

XMITIPED is an ISPF Edit command that will take the data set name (and member) and invoke the XMITIPFE command to bring up the XMITIP ISPF dialog.

Syntax: XMITIPED

## XMITIPFE

This tool is to simplify calling the XMITIP ISPF dialog from other ISPF applications. It allows the calling application to pass information that will be set in the XMITIP ISPF panel for the user. The syntax is:

Syntax: %xmitipfe keywords

Valid keywords:

- To(to-email-address)
- msgds(\* for edit or data set name)
- file(attachment data set name)
- filename(attachment name)
- format(format)
- subject(subject text)

## XMITIPPD

This exec now invokes TXT2PDF.

## XMITBULK

This exec allows the user to create a sequential file with many different e-mails within it. The syntax is:

%xmitbulk bulk-dsn FROM from-addr options

where **bulk-dsn** is a sequential dataset with contents described below or

dd:ddname which refers to a sequential dataset with contents described below

**From-addr** is the sender e-mail address (e.g. FROM [your.name@company.com](mailto:your.name@company.com))

**Options:** **HTML** which will send the message as enriched text

**CONFIG** *dsn* or **CONFIGDD** *ddname* to reference a XMITIP configuration file or ddname

Bulk-DSN data format starting in column 1:

TO: to-address

CC: cc-address

SUBJECT: subject

Text

Text

Only 1 TO and 1 SUBJECT are allowed.

## Reading the Attachment on the Workstation:

When reading the attachment in your e-mail reader the user can either (1) launch the attachment, which will open the attachment in the appropriate application or (2) if you are using Lotus Notes you can try using the built in Lotus Notes View option. When using Microsoft Outlook when clicking on the attachment you will be asked to open or save it. When you select Open the attachment is opened in the appropriate application.

Note: If the attachment is a text file you may need to change the default font as the default font for text file is normally a proportional font while mainframe reports tend to be designed to use a fixed pitch font (e.g. Courier).

In Microsoft NOTEPAD to change the default font:

- Select EDIT
- Select Set Font
- Select Courier or Courier New
- Select a reasonable font size (e.g. 10)

## Change History

For a complete list of changes see the installation dataset member CHANGES.