

# Unicode Services

## Batch Tools for z/OS

User Manual  
Version 1

Copyright IBM Corporation 2016

## Contents

1. Overview .....	3
2. CUNMTUNI: Unicode Services Batch Client .....	4
2.1 Introduction .....	4
2.2 Example .....	4
2.3 User Interface .....	5
3. CUNMCSMX/CUNMRCSX: sending MIME data via SMTP .....	8
3.1 Introduction .....	8
3.2 Send Mail from the Batch .....	8
3.2.1 Example .....	9
3.2.3 User Interface .....	9
3.2.4 Return Codes .....	11
3.3 Send Mail from COBOL Program .....	11
4. CUNMCSMH/CUNMRCSH: sending HTML data via SMTP .....	15
4.1 Introduction .....	15
4.2 Send Mail from the Batch .....	15
4.2.1 Example .....	16
4.2.2 User Interface .....	16
4.2.3 Return Codes .....	18
4.3 Send Mail from COBOL Program .....	18
5. CUNMCSMM/CUNMRCSM: sending MIME data via SMTP .....	22
5.1 Introduction .....	22
5.2 Send Mail from the Batch .....	22
5.2.1 Enhancements compared with CUNMRCSX .....	23
5.2.2 Example .....	24
5.2.3 User Interface .....	25
5.2.4 Comments in PARMDD .....	30
5.2.5 Return Codes .....	30
5.2.6 Other Examples .....	31
5.3 Send Mail from COBOL Program .....	34

# 1. Overview

Unicode Services now provides some tools to allow users in a batch environment to:

- 1) Access the different Unicode Services provided.
- 2) Send MIME data to mail recipients.
- 3) Send HTML embedded multimedia data to mail recipients.

The rest of this document describes how to use this support.

# 2.CUNMTUNI:

## Unicode Services Batch Client

### 2.1 Introduction

The intention of this program is to use the functions provided by the z/OS Unicode Services in batch processing. These functions are described in detail in “Unicode Services User’s Guide and Reference”.

The module name of this program is ‘CUNMTUNI’.

The module runs in 64-bit mode.

### 2.2 Example

This example shows how to invoke one of the supported functions, Normalization, with CUNMTUNI.

```
//STEPEXEC PGM=CUNMTUNI
//PARMFILE DD *
*request function 'normalization'
FUNC=N
*ask for type NFC
TYPE=NFC
*input is code page EBCDIC Austrian/German
CCID=1141
*insert X'0D' after reading of an input record
CRLF=1
*insert Byte Order Mark X'FEFF' to indicate UTF-16 BE
UBOM=0
/*
//DATIN DD DISP=SHR, DSN=HLQ.DS.INPUT
//DATOUT DD DISP=SHR, DSN=HLQ.DS.OUTPUT
//SYSOUT DD SYSOUT=*
```

## 2.3 User Interface

### **PARMFILE: DD**

Description: Input data set containing the driving parameters to CUNMTUNI that represents the Unicode functions to be invoked as well as other parameters related with that Unicode function.

Parameters:

**FUNC:** Unicode function indicator.

- N Normalization
- A Case Conversion
- S Stringprep
- C Collation
- V Character Conversion (incl. Bidirectional)

**TYPE:** Type of specific Unicode Service. Its possible values vary with the Unicode Service, namely the value of FUNC. And it is available when FUNC=N/A/S/C.

FUNC=N:

- NFC Canonical composition
- NFD Canonical decomposition
- NFKC Compatibility decomposition
- NFKD Compatibility decomposition

FUNC=A:

- T Conversion to title case.
- L Conversion to lower case.
- U Conversion to upper case.

FUNC=S:

- UTF-8 Check if input text is in UTF-8 representation.
- UTF-16 Check if input text is in UTF-16 representation.

FUNC=C:

- <locale> Locale for the collation.

**CCID:** Input data CCSID indicator. It is used to convert the input data. If missing, the character conversion will not be performed.

**TCID:** Output data CCSID indicator. It indicates the CCSID of the target data converted from input data in character conversion. If missing, it will be set as 1200, representing the latest Unicode version.

**CRLF:** Indicates the value to be inserted after reading an input record or XML data is to be processed or variable format output data set is to be used.

- 0: Used for EBCDIC data, and X'15' will be appended after reading a record.
- 1: Used for ASCII data, and X'0D' will be appended after reading a record.
- 2: Used for ASCII data, and X'0D0A' will be appended after reading a record.
- S: XML data is to be processed. All end line characters will be removed and line is split when LRECL is exceeded or a new XML statement appears.
- V: Used for EBCDIC data, and X'25' will be appended after reading a record. And variable format output data set will be used. Record is written according to the value of RECL which must be specified when CRLF=V.

**RECL:** Indicates the amount of bytes of one record in output data set. Valid only when CRLF=V.

**UBOM:** Byte Order Mask, indicates the value to be inserted before writing an output record.

- 0: X'FEFF' UTF-16 BE
- 1: X'EFBBBF' UTF-8

**BKEY:** Used when FUNC=V, indicates extended BIDI service is to be invoked. For more information on possible BKEY values please refer to Unicode Services User's Guide and Reference.

**NOTE:** Character conversion is performed automatically if CCID specified, and 1200 will be the default target when you don't specify TCID.

**NOTE:** In Normalization and Case Conversion, either specifying CCID for input data or skipping parameter CCID to have the input encoded in 1200.

**NOTE:** The length of TYPE value must be less than or equal to 8.

**NOTE:** For Collation Service, DATOUT DD must be set as DUMMY.

**NOTE:** PARMFILE is usually used as an in-stream data set, otherwise, its attributes must be set as DSORG=PS, RECFM=FB, LRECL=80.

**DATIN: DD**

Description: Input data set containing the data to be processed by Unicode Services.

NOTE: DATIN can be of any RECFM, and there is no limit on its LRECL.

**DATOUT: DD**

Description: Output data set containing the processed result of input data from Unicode Service.

NOTE: For different CRLF values, DATOUT must be in different attributes as follow, or unexpected results would occur.

If CRLF=S, DATOUT attributes must be DSORG=PS, RECFM=VB, LRECL=19996, BLKSIZE=20000;

If CRLF=V, they must be DSORG=PS, RECFM=V, LRECL=200;

Otherwise, they must be DSORG=PS, RECFM=U.

**SYSOUT: DD**

Description: Output data set which will contain the message generated from checking the parameters in PARMFWILE.

NOTE: It is usually set as SYSOUT=\*, otherwise, its attributes must be DSORG=PS, RECFM=FB, BLKSIZE=13300, LRECL=80.

# 3. CUNMCSMX/CUNMRCSX: sending MIME data via SMTP

## 3.1 Introduction

The intention of this program is to send MIME data via SMTP. It will write user data in SMTP format into a JES-SPOOL dataset. CSSMTP will handle the dataset and send it as a MIME message.

The module name of this program is 'CUNMRCSX/CUNMCSMX'.

You can both send them in a program using the CUNMCSMX or from the batch interface CUNMRCSX.

## 3.2 Send Mail from the Batch

CUNMRCSX provides a solution to send user data to mail recipients from the batch. You can send multiple MIME parts in one message to the recipients as text or binary attachments:

- EBCDIC Text inline or as attachments.
- EBCDIC HTML files inline or as attachments.
- Binary files as attachments.

The matrix below depicts the usable formats:

Input	Keyword	Attachment name	Output
EBCDIC text	TXT	-	Inline message
EBCDIC text	TXT	yes	'xxxx.txt' or csv, rtf.
Binary	BIN	yes	pdf, doc, xls, jpg.
EBCDIC text	BINU	-	Inline UTF-16
EBCDIC text	BINU	yes	'xxxx.txt' or csv, rtf.
EBCDIC text	BIN8	-	Inline UTF-8
EBCDIC text	BIN8	yes	'xxxx.txt' or csv, rtf.
EBCDIC HTML	HTM	-	Inline HTML
EBCDIC HTML	HTM	yes	'xxxx.htm'

If you send binary data, make sure to have a format that the recipient is able to read. That can be any binary data e.g. a pdf or jpg. Be aware that some file types may be suppressed by the recipient's mailing software.



## 3.2.1 Example

Here is an example JCL of using CUNMRCSX:

```
//STEP      EXEC   PGM=CUNMRCSX
//MIMOUT      DD   SYSOUT=(A,SMTP)
//SNAPDUMP    DD   DUMMY      --if needed SYSOUT=*--
//PARMDD      DD   *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=TEST MAIL OF CUNMRCSX
ORIG=SENDER@IBM.COM
RECP=RECPT@IBM.COM
DATA=DATA1DD HTM
DATA=DATA2DD TXT
//DATA1DD DD   DISP=SHR, DSN=HLQ.CNTL (DATA1DD)
//DATA2DD DD   DISP=SHR, DSN=HLQ.CNTL (DATA2DD)
```

### Remarks:

You have to specify REGION=0M in your JOB or EXEC card.

## 3.2.3 User Interface

### MIMOUT: DD

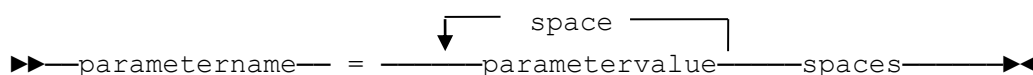
Description: Output DD containing MIME data which will be processed by SMTP application and sent to the recipients. E.g. SYSOUT = (A, < CSSMTP JOB NAME >).

### SNAPDUMP: DD

Description: Trace DD, in case requested, please do SYSOUT=\*, otherwise keep dummy to avoid diagnostics.

### PARMDD: DD

Description: You code all parameters in this statement. The syntax follows.



- HOST** Defines HOSTNAME, required but any value can be specified.
- EHLO** Default SMTP introduction clause, preferred for CSSMTP.
- HELO** Defines SMTP HELO clause as HELO. (E.g. old SMTPD).  
(You need not specify those parameters for CSSMTP, EHLO is automatically added).
- TRAC** Trace indicator.  
Parameter can be omitted, setting default no trace.

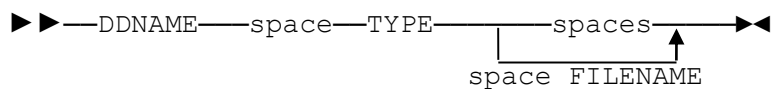
If present must be second parameter.

Possible values if present

- 0 no trace (default)
- 8 maximal trace
- 9 extended trace (user data)

**SUBJ** Subject parameter. Taken as is till up to 109 characters if available.  
**ORIG** Sender of a mail message (maximum length 69). Must precede DATA.  
**RECP** Recipient of a mail message (maximum length 69). Must precede DATA or MESH.  
 For multiple recipients, specify multiple RECP statements.  
**DATA** Describes data to be sent. For multiple bodies, code multiple DATA statements.

Syntax:



**DDNAME** DD name you must provide in JCL as datandd according to example JCL.

**TYPE** Corresponds to XTSONL-DATA-MIME-TYP COBOL parameter. For details please check following table.

Value	Meaning
TXT	EBCDIC (1141) input, ASCII (923) output, CRLF added between lines. Typically output from COBOL program or text file.
HTM	EBCDIC (1141) input, ASCII (923) output.
BIN	Input is binary data, no data conversion is done, Typically encrypted data, or pre-composed hex values.
BIN <sub>x</sub> nnnn	nnnn stands for input CCSID, in case not present defaulted to CCSID 1141.
x=U	Conversion to UTF-16 is performed upon input data.
x=8	Conversion to UTF-8 is performed upon input data.
	Example: BINU1141 to convert text from CCSID 1141 to UTF-16.

**FILENAME** If present, data will be interpreted as attachment and FILENAME will be used as name for this attachment. In case FILENAME is omitted, data is interpreted as inline text.

## 3.2.4 Return Codes

0	Normal exit
4	Exit with warnings
8	Invalid parameter keyword or parameter length
C	Internal error

## 3.3 Send Mail from COBOL Program

You can also put your data directly into a COBOL program (call CUNMCSMX module) to send user data to mail recipients.

Description and declaration in COBOL format:

```
*****
*
*       C O P Y - M E M B E R F O R       C U N M C S M X       *
*
*****
01 XTSONL-RUECKPARM.
   *--- VALUE 0078 at entry to request HELO in SMTP protocol
05 XTSONL-RUECKPARM-LG          PIC 9(04)  VALUE 0077.
05 XTSONL-RETURN-CODE          PIC 9(03)  VALUE ZERO.
   *--- successful run
   88 XTSONL-RET-OK              VALUE ZERO.
   *--- wrong parameters
   88 XTSONL-RET-UEB             VALUE 008.
   *--- error in sub call
   88 XTSONL-RET-VERARB         VALUE 012.
   *--- error text
05 XTSONL-RETURN-TXT           PIC X(73)  VALUE SPACE.

01 XTSONL-ALLPARM.
   *--- This parameter represents the server host name
   *--- containing the mail server functions.
03 XTSONL-HOSTPARM.
   *--- Length of entire parameter
05 XTSONL-HOSTPARM-LG          PIC 9(05)  VALUE 00149.
   *--- Argument for sub call
05 XTSONL-HOSTPARM-TYP         PIC X(08)  VALUE 'HOSTPARM'.
   *--- PORT, leave empty
05 XTSONL-HOSTPARM-PORT        PIC 9(05)  VALUE ZERO.
   *--- Length of HOST parameter
05 XTSONL-HOSTPARM-HOST-LG     PIC 9(03)  VALUE ZERO.
```

```

*--- HOST parameter
05 XTSONL-HOSTPARM-HOST          PIC X(128)  VALUE SPACE.

*--- This parameter contains the TRACE flag
03 XTSONL-UEBPARMF.
  *--- Length of entire parameter
  05 XTSONL-UEBPARMF-LG          PIC 9(05)   VALUE 00014.
  *--- Argument for sub call
  05 XTSONL-UEBPARMF-TYP        PIC X(08)   VALUE 'UEBPARMF '.
  *--- Trace level 0 or 8
  05 XTSONL-UEBPARMF-TRACE-LVL  PIC 9(01)   VALUE ZERO.
      88 XTSONL-KEIN-TRACE          VALUE ZERO.
      88 XTSONL-MAX-TRACE          VALUE 8.
      88 XTSONL-EXT-TRACE          VALUE 9.

*--- This parameter group contains a number of fixed and
*--- variable times callable parameters, specifying the
*--- appropriate arguments for subject, originator and
*--- recipient(s)
*--- UEBPARMV reserved max length 6800          (variable)
*--- Header incl. entire length: 13            (variable)
*--- Subj length incl. header and length 120    (fixed)
*--- ORIG length incl. header and length 80     (fixed)
*--- RECP length incl. header and length 80
*--- (max 80 recps, each fixed length 80 = max 6400)
03 XTSONL-UEBPARMV.
  *--- Length of entire parameter set used (max 6800)
  05 XTSONL-UEBPARMV-LG          PIC 9(05)   VALUE ZERO.
  *--- Argument for sub call
  05 XTSONL-UEBPARMV-TYP        PIC X(08)   VALUE 'UEBPARMV '.
  *--- Subject parameter fixed length 120
  05 XTSONL-UEBPARMV-TYP-S
      07 XTSONL-SUBJECT          PIC X(08)   VALUE 'SUBJECT '.
      07 XTSONL-SUBJECT-LG       PIC 9(03)   VALUE ZERO.
      07 XTSONL-SUBJECT-FIELD    PIC X(109)  VALUE SPACE.
  *--- Originator parameter fixed length 80
  05 XTSONL-UEBPARMV-TYP-O
      07 XTSONL-ORIG             PIC X(08)   VALUE 'FROM '.
      07 XTSONL-ORIG-LG         PIC 9(03)   VALUE ZERO.
      07 XTSONL-ORIG-FIELD      PIC X(69)   VALUE SPACE.
  *--- Recipient parameter each fixed length 80,
  *--- max 80 recipients when recipient list
  05 XTSONL-UEBPARMV-TYP-R
      07 XTSONL-RECP            PIC X(08)   VALUE SPACE.

```

```

      88 XTSONL-RECP-TO          VALUE 'TO      '.
07 XTSONL-RECP-LG             PIC 9(03)  VALUE ZERO.
07 XTSONL-RECP-FIELD         PIC X(69)   VALUE SPACE.

*--- This parameter is for mail data.
03 XTSONL-DATA.
  *--- Length of entire DATA parameter
  *--- It is not evaluated. User can put there any value
  *--- for documentation, but it must be 8 digits.
  *--- If not used, specify 8 zeros.
05 XTSONL-DATA-LG             PIC 9(08) .
  *--- Argument for sub call
05 XTSONL-DATA-TYP           PIC X(08)   VALUE 'DATA  '.
  *--- Mark (E), if this is last data body part. Default
  *--- Mark (M), if there are additional body parts.
05 XTSONL-DATA-FLAG          PIC X(01)   VALUE SPACE.
      88 XTSONL-DATA-END          VALUE 'E'.
      88 XTSONL-DATA-MORE        VALUE 'M'.
  *--- Mark if attachment.
05 XTSONL-DATA-ATT-FLAG      PIC X(01)   VALUE SPACE.
      88 XTSONL-DATA-ATTACHMENT  VALUE 'A'.
      88 XTSONL-DATA-TEXT        VALUE 'T'.
  *--- Attachment name
05 XTSONL-DATA-ATTNAME       PIC X(64)   VALUE SPACE.
  *--- For embedded files MIME type only:
      07 XTSONL-DATA-ATTNAME-CONTENT-TYPE PIC X(24) .
      07 XTSONL-DATA-ATTNAME-LOCATION    PIC X(14) .
  *--- MIME type
  *--- TXT    for Text data, default.
  *--- HTM    for HTML data.
  *--- BINU   for Text data.
  *--- BIN8   for Text data.
  *--- BIN    for Binary data.
  *--- No character conversion will be done for Binary data.
  *--- Both HTML and Text data will be converted accordingly.
  *--- HTM/TXT is converted from Default EBCDIC to default ASCII.
  *--- BINU is converted from MIME-TYP-CCSID to UTF-16.
  *--- BINU is converted from MIME-TYP-CCSID to UTF-8.
05 XTSONL-DATA-MIME-TYP      PIC X(64)   VALUE SPACE.
      07 XTSONL-DATA-MIME-TYP-MIME    PIC X(03) .
      07 XTSONL-DATA-MIME-TYP-UNI     PIC X.
          88 XTSONL-DATA-MIME-TYP-UNIUTF-16 VALUE 'U'.
          88 XTSONL-DATA-MIME-TYP-UNIUTF-8  VALUE '8'.
      07 XTSONL-DATA-MIME-TYP-CCSID    PIC X(04) VALUE 1141.

```

```

07 XTSONL-DATA-MIME-FILLER      PIC X(56)      VALUE SPACE.
*--- Length of data (theoretical max 99999999)
*--- Minimum size in binary files 40 bytes,
*--- else unpredictable results
05 XTSONL-DATA-STRING-LG      PIC 9(08)              VALUE ZERO.
*--- Data contents
05 XTSONL-DATA-STRING          PIC X(max. 99999999) VALUE SPACE.

```

### Detail Description:

#### Reg. RECP and ORIG:

Please specify real length so no blanks are generated. Even if the declaration of the various address fields are 69 characters, you should specify the real length in the leading length field (e.g. XTSONL-RECP-LG).

#### Reg. SUBJ:

The SUBJ value is taken as it is, you can specify anything in the 109 characters, even blanks. Its length is a comment only and not evaluated. It is recommended to specify a value of 109.

#### Reg. DATA:

You can code up to 100 data blocks.

XTSONL-DATA-FLAG must be 'E' in last block of chain.

Specifying 'M' in last DATA block may cause unpredictable results.

The multiple body parts may be mixed either as binary or text, inline or attachment. Binary data should be specified as an attachment, because the capability of the receiving mail server may not support binary data in an inline format.

#### Reg. UNICODE:

Specify 'BIN' as XTSONL-DATA-MIME-TYP-MIME,  
use 'U' as XTSONL-DATA-MIME-TYP-UNI for UTF-16,  
use '8' as XTSONL-DATA-MIME-TYP-UNI for UTF-8.

### Coding the user program:

- The entry point being called is 'CUNMCSMX'.
- Please take a valid 'FROM' address.
- The '@' sign may require a substitution character like '§' in some national code pages. Please refer to the CODEPAGE value in CSSMTP.
- To indicate a new line within the data part,
  - ✓ code X'0D', for XTSONL-DATA-MIME-TYP=TXT,
  - ✓ code X'0D15' for XTSONL-DATA-MIME-TYP=BIN8 and XTSONL-DATA-MIME-TYP=BINU.

(In batch program CUNMRCSH this is done automatically)

# 4. CUNMCSMH/CUNMRCSH: sending HTML data via SMTP

## 4.1 Introduction

The intention of this program is to send HTML embedded multimedia data via SMTP. The result of the program is to write user data in SMTP format into JES-SPOOL dataset. CSSMTP will handle the dataset and send it as a MIME message.

The module name of this program is 'CUNMRCSH/CUNMCSMH'.

You can both sending them in a program using the CUNMCSMH or from the batch interface CUNMRCSH.

## 4.2 Send Mail from the Batch

CUNMRCSH provides a solution to send HTML embedded multimedia data to mail recipients from the batch.

You can send multiple MIME parts in one message to the recipients as text attachments or inline parts:

- EBCDIC HTML inline (must be the first MIME part).
- EBCDIC HTML as attachments.
- EBCDIC Text as attachments.
- Binary inline files.

The matrix below depicts the usable formats:

Input	Keyword	Attachment name	Output
EBCDIC text	TXT	Yes	'xxxx.txt' or csv, rtf.
Binary	BIN	-	HTML embedded resource.
EBCDIC text	BINU	Yes	'xxxx.txt' or csv, rtf.
EBCDIC text	BIN8	Yes	'xxxx.txt' or csv, rtf.
EBCDIC HTML	HTM	-	Inline HTML

If you send binary data, make sure to have a format that the recipient is able to read. That can be any binary data e.g. a pdf or jpg. Be aware that some file types may be suppressed by the recipient's mailing software.

## 4.2.1 Example

Here is an example JCL of using CUNMRCSH:

```
//STEP      EXEC PGM=CUNMRCSH
//MIMOUT    DD SYSOUT=(A,SMTP)
//SNAPDUMP  DD DUMMY      --if needed SYSOUT=*--
//PARMDD    DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=TEST MAIL OF CUNMRCSH
ORIG=SENDER@IBM.COM
RECP=RECPT@IBM.COM
DATA=DATA1DD HTM
DATA=DATA2DD BIN IMAGE/JPEDG IMAGEID
//DATA1DD DD DISP=SHR,DSN=HLQ.CNTL(DATA1DD)
//DATA2DD DD DISP=SHR,DSN=HLQ.CNTL(DATA2DD)
```

### Remarks:

You have to specify REGION=0M in your JOB or EXEC card.

## 4.2.2 User Interface

### MIMOUT: DD

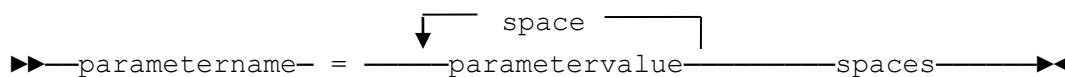
Description: Output DD containing MIME data which will be processed by SMTP application and sent to the recipients. E.g. SYSOUT = (A, < CSSMTP JOB NAME>).

### SNAPDUMP: DD

Description: Trace DD, in case requested, please do SYSOUT=\*, otherwise keep dummy to avoid diagnostics.

### PARMDD: DD

Description: You code all parameters in this statement. The syntax follows.



- HOST** Defines HOSTNAME, required but any value can be specified.
- EHLO** Default SMTP introduction clause, preferred for CSSMTP.
- HELO** Defines SMTP HELO clause as HELO. (e.g. old SMTPD).  
(You need not specify those parameters for CSSMTP, EHLO is automatically added).
- TRAC** Trace indicator.  
Parameter can be omitted, setting default no trace.



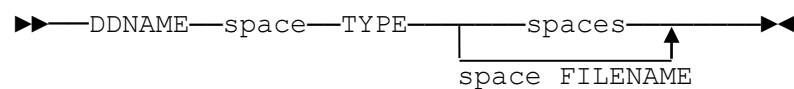
If present must be second parameter.

Possible values if present

- 0 no trace (default)
- 8 maximal trace
- 9 extended trace (user data)

**SUBJ** Subject parameter. Taken as is till up to 109 characters if available.  
**ORIG** Sender of a mail message (maximum length 69). Must precede DATA.  
**RECP** Recipient of a mail message (maximum length 69). Must precede DATA or MESH.  
 For multiple recipients, specify multiple RECP statements.  
**DATA** Describes data to be sent. For multiple bodies, code multiple DATA statements.

Syntax:



**DDNAME** DD name you must provide in JCL as datandd according example JCL.

**TYPE** Corresponds to XTSONL-DATA-MIME-TYP COBOL parameter. For details please check following table.

Value	Meaning
TXT	EBCDIC (1141) input, ASCII (923) output, CRLF added between lines. Typically output from COBOL program or text file.
HTM	EBCDIC (1141) input, ASCII (923) output.
BIN	Input is binary data, no data conversion is done, Typically encrypted data, or pre-composed hex values.
BINxnnnn	nnnn stands for input CCSID, in case not present defaulted to CCSID 1141.
x=U	Conversion to UTF-16 is performed upon input data.
x=8	Conversion to UTF-8 is performed upon input data.
	Example: BINU1141 to convert text from CCSID 1141 to UTF-16.

**FILENAME** If present, data will be interpreted as attachment and FILENAME will be used as name for this attachment. In case FILENAME is omitted, data is interpreted as inline text.

## 4.2.3 Return Codes

0	Normal exit
4	Exit with warnings
8	Invalid parameter keyword or parameter length
C	Internal error

## 4.3 Send Mail from COBOL Program

You can also put your data directly into a COBOL program (call to CUNMCSMH module) to send user data to mail recipients.

Description and declaration in COBOL format:

```
*****
*
*          C O P Y - M E M B E R F O R          C U N M C S M H      *
*
*****
01 XTSONL-RUECKPARAM.
   *--- VALUE 0078 at entry to request HELO in SMTP protocol
05 XTSONL-RUECKPARAM-LG          PIC 9(04)  VALUE 0077.
05 XTSONL-RETURN-CODE          PIC 9(03)  VALUE ZERO.
   *--- successful run
   88 XTSONL-RET-OK              VALUE ZERO.
   *--- wrong parameters
   88 XTSONL-RET-UEB             VALUE 008.
   *--- error in sub call
   88 XTSONL-RET-VERARB          VALUE 012.
   *--- error text
05 XTSONL-RETURN-TXT            PIC X(73)  VALUE SPACE.

01 XTSONL-ALLPARAM.
   *--- This parameter represents the server host name
   *--- containing the mail server functions.
03 XTSONL-HOSTPARAM.
   *--- Length of entire parameter
05 XTSONL-HOSTPARAM-LG          PIC 9(05)  VALUE 00149.
   *--- Argument for sub call
05 XTSONL-HOSTPARAM-TYP          PIC X(08)  VALUE 'HOSTPARAM'.
   *--- PORT, leave empty
05 XTSONL-HOSTPARAM-PORT          PIC 9(05)  VALUE ZERO.
   *--- Length of HOST parameter
05 XTSONL-HOSTPARAM-HOST-LG      PIC 9(03)  VALUE ZERO.
```

```

*--- HOST parameter
05 XTSONL-HOSTPARM-HOST          PIC X(128)  VALUE SPACE.

*--- This parameter contains the TRACE flag
03 XTSONL-UEBPARMF.
*--- Length of entire parameter
05 XTSONL-UEBPARMF-LG            PIC 9(05)   VALUE 00014.
*--- Argument for sub call
05 XTSONL-UEBPARMF-TYP          PIC X(08)   VALUE 'UEBPARMF '.
*--- Trace level 0 or 8
05 XTSONL-UEBPARMF-TRACE-LVL    PIC 9(01)   VALUE ZERO.
    88 XTSONL-KEIN-TRACE                VALUE ZERO.
    88 XTSONL-MAX-TRACE                VALUE 8.
    88 XTSONL-EXT-TRACE                VALUE 9.

*--- This parameter group contains a number of fixed and
*--- variable times callable parameters, specifying the
*--- appropriate arguments for subject, originator and
*--- recipient(s)
*--- UEBPARMV reserved max length 6800                (variable)
*--- Header incl. entire length: 13                    (variable)
*--- Subj length incl. header and length 120           (fixed)
*--- ORIG length incl. header and length 80            (fixed)
*--- RECP length incl. header and length 80
*--- (max 80 recps, each fixed Length 80 = max 6400)
03 XTSONL-UEBPARMV.
*--- Length of entire parameter set used (max 6800)
05 XTSONL-UEBPARMV-LG            PIC 9(05)   VALUE ZERO.
*--- Argument for sub call
05 XTSONL-UEBPARMV-TYP          PIC X(08)   VALUE 'UEBPARMV '.
*--- Subject parameter fixed length 120
05 XTSONL-UEBPARMV-TYP-S
    07 XTSONL-SUBJECT            PIC X(08)   VALUE 'SUBJECT  '.
    07 XTSONL-SUBJECT-LG         PIC 9(03)   VALUE ZERO.
    07 XTSONL-SUBJECT-FIELD      PIC X(109)  VALUE SPACE.
*--- Originator parameter fixed length 80
05 XTSONL-UEBPARMV-TYP-O
    07 XTSONL-ORIG              PIC X(08)   VALUE 'FROM  '.
    07 XTSONL-ORIG-LG           PIC 9(03)   VALUE ZERO.
    07 XTSONL-ORIG-FIELD        PIC X(69)   VALUE SPACE.
*--- Recipient parameter each fixed length 80,
*--- max 80 recipients when recipient list
05 XTSONL-UEBPARMV-TYP-R
    07 XTSONL-RECP              PIC X(08)   VALUE SPACE.

```

```

      88 XTSONL-RECP-TO                VALUE 'TO      '.
07 XTSONL-RECP-LG                    PIC 9(03)  VALUE ZERO.
07 XTSONL-RECP-FIELD                 PIC X(69)  VALUE SPACE.

*--- This parameter is for mail data.
03 XTSONL-DATA.
  *--- Length of entire DATA parameter
  *--- It is not evaluated. User can put there any value
  *--- for documentation, but it must be 8 digits.
  *--- If not used, specify 8 zeros.
05 XTSONL-DATA-LG                    PIC 9(08) .
  *--- Argument for sub call
05 XTSONL-DATA-TYP                  PIC X(08)  VALUE 'DATA  '.
  *--- Mark (E), if this is last data body part. Default
  *--- Mark (M), if there are additional body parts.
05 XTSONL-DATA-FLAG                 PIC X(01)  VALUE SPACE.
      88 XTSONL-DATA-END                VALUE 'E'.
      88 XTSONL-DATA-MORE               VALUE 'M'.
  *--- Mark if attachment.
05 XTSONL-DATA-ATT-FLAG             PIC X(01)  VALUE SPACE.
      88 XTSONL-DATA-ATTACHMENT        VALUE 'A'.
      88 XTSONL-DATA-TEXT              VALUE 'T'.
  *--- Attachment name
05 XTSONL-DATA-ATTNAME              PIC X(64)  VALUE SPACE.
  *--- For embedded files MIME type only:
      07 XTSONL-DATA-ATTNAME-CONTENT-TYPE PIC X(24) .
      07 XTSONL-DATA-ATTNAME-LOCATION    PIC X(14) .
  *--- MIME type
  *--- TXT    for Text data, default.
  *--- HTM    for HTML data.
  *--- BINU   for Text data.
  *--- BIN8   for Text data.
  *--- BIN    for Binary data.
  *--- No character conversion will be done for Binary data.
  *--- Both HTML and Text data will be converted accordingly.
  *--- HTM/TXT is converted from Default EBCDIC to default ASCII.
  *--- BINU is converted from MIME-TYP-CCSID to UTF-16.
  *--- BINU is converted from MIME-TYP-CCSID to UTF-8.
05 XTSONL-DATA-MIME-TYP             PIC X(64)  VALUE SPACE.
      07 XTSONL-DATA-MIME-TYP-MIME      PIC X(03) .
      07 XTSONL-DATA-MIME-TYP-UNI      PIC X.
          88 XTSONL-DATA-MIME-TYP-UNIUTF-16 VALUE 'U'.
          88 XTSONL-DATA-MIME-TYP-UNIUTF-8  VALUE '8'.
      07 XTSONL-DATA-MIME-TYP-CCSID    PIC X(04)  VALUE 1141.

```

```

07 XTSONL-DATA-MIME-FILLER      PIC X(56)      VALUE SPACE.
*--- Length of data (theoretical max 99999999)
*--- Minimum size in binary files 40 bytes,
*--- else unpredictable results
05 XTSONL-DATA-STRING-LG      PIC 9(08)      VALUE ZERO.
*--- Data contents
05 XTSONL-DATA-STRING          PIC X(max. 99999999)  VALUE SPACE.

```

### Detail description:

#### Reg. RECP and ORIG:

Please specify real length so no blanks are generated. Even if the declaration of the various address fields are 69 characters, you should specify the real length in the leading length field (e.g. XTSONL-RECP-LG).

#### Reg. SUBJ:

The SUBJ value is taken as it is, you can specify anything in the 109 characters, even blanks. Its length is a comment only and not evaluated. It is recommended to specify a value of 109.

#### Reg. DATA:

You can code up to 100 data blocks.

XTSONL-DATA-FLAG must be 'E' in last block of chain.

Specifying 'M' in last DATA block may cause unpredictable results.

The multiple body parts may be mixed either as binary or text, inline or attachment.

#### Reg. UNICODE:

Specify 'BIN' as XTSONL-DATA-MIME-TYP-MIME,

use 'U' as XTSONL-DATA-MIME-TYP-UNI for UTF-16,

use '8' as XTSONL-DATA-MIME-TYP-UNI for UTF-8.

### Coding the user program:

- The entry point being called is 'CUNMCSMH'.
- Please take a valid 'FROM' address.
- The '@' sign may require a substitution character like '§' in some national code pages. Please refer to the CODEPAGE value in CSSMTP.
- To indicate a new line within the data part,
  - ✓ code X'0D', for XTSONL-DATA-MIME-TYP=TXT,
  - ✓ code X'0D15' for XTSONL-DATA-MIME-TYP=BIN8 and XTSONL-DATA-MIME-TYP=BINU.

(In batch program CUNMRCSH this is done automatically.)

# 5. CUNMCSMM/CUNMRCSM: sending MIME data via SMTP

## 5.1 Introduction

The intention of this tool is to send MIME data via SMTP. The result of the program is a MIMEOUT file, which is to be put into the JES-SPOOL. CSSMTP will handle the file and send it as an SMTP message.

The module name of this program is 'CUNMRCSM/CUNMCSMM'.

You can both send them in a program using the CUNMCSMM or from the batch interface CUNMRCSM. CUNMRCSM is an enhanced version of CUNMRCSX, which provides more capabilities.

## 5.2 Send Mail from the Batch

CUNMRCSM provides a solution to send more kinds of MIME data to mail recipients from the batch compared with previous CUNMRCSX.

You can send multiple MIME parts in one message to the recipients as text attachments or inline parts:

- EBCDIC HTML inline or as attachments.
- EBCDIC Text inline or as attachments.
- Binary data embedded in HTML or as attachments.
- Binary P7M data as attachments
- Binary P7S data as attachments

The matrix below depicts the usable formats:

Input	Keyword	Attachment name	Output
EBCDIC text	TXT	-	Inline message
EBCDIC text	TXT	Yes	Text attachment file
Binary	BIN	Yes	Binary attachment file
EBCDIC text	BINU	-	Inline UTF-16 text file
EBCDIC text	BINU	Yes	UTF-16 attachment file
EBCDIC text	BIN8	-	Inline UTF-8 text file
EBCDIC text	BIN8	Yes	UTF-8 attachment file
EBCDIC text	BINA	-	Inline ASCII text file

Input	Keyword	Attachment name	Output
EBCDIC text	BINA	Yes	ASCII attachment file
Binary	P7M	Yes	P7M file enveloped
Binary	P7S	Yes	P7S file signed
HTML text followed by object	CID blocks: CID CID:<multimedia type> <content location specified in CID HTML body>	-	The first 'CID' points HTML text. Subsequent 'CID' points to the embedded objects.

If you send binary data, make sure to have a format that the recipient is able to read. That can be any binary data e.g. a pdf or jpg file. Be aware that some file types may be suppressed by the recipient's mailing software.

## 5.2.1 Enhancements compared with CUNMRCSX

- Refer to DD: DATA=&<ddname>, optionally followed by <type> and <attachment name>.
- Same for RECP, COPY and BCPY, not for ORIG!
- CONFIGDD DD statement is added to allow users to configure the default EBCDIC and ASCII CCSIDs.
- MMSG, for short messages, max 80 bytes, you can include some system variables (&DATE, &TIME, &LPAR, &USER, &JOB, &TIM0).
- CRLF parameter to insert an additional new line feed after MMSG or DATA, unless being performed by the receiving mail system. Specify CRLF=2 before MMSG/DATA parameters in PARMDD.
- Syntax in CID statement: CID (starting from second CID) is followed by ':' and <multimedia type>, and finally by <file name>.  
Sample: DATA=my.dsn CID:image/jpeg myimage.jpg.
- ATXL parameter to translate old X'7C' national substitution characters (§, Ö) to '@'. ATXL must be specified before ORIG/RECP parameters in PARMDD.
- System variables &LPAR, &DATE, &TIME, &USER, &JOB, &JNUM, &TIM0 can be included in HOST, MMSG, SUBJ and attachment name.
- BINA parameter: text is translated according to UNICODE services from specified EBCDIC code page to default ASCII code page and is sent binary in base64 format.
- Optional use of full name using the "name" <mail address> format for ORIG and SEND.
- New parameter RPLY for the Reply-To function.
- New parameter SEND for the Sender function.
- New parameter MAXM for the MAX MESSAGE SIZE function.
- New parameter VARI to set variable prefix character instead of fixed '&'.
- CRLF parameter is extended.
- Comments in parameters are changed.

## 5.2.2 Example

Here is a JCL example using CUNMRCSM:

```
//STEP      EXEC PGM=CUNMRCSM
//MIMOUT    DD SYSOUT=(A, SMTP)
//SNAPDUMP DD DUMMY      --if needed SYSOUT=*--
//PARMDD    DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=TEST MAIL OF CUNMRCSM
ORIG=SENDER@IBM.COM
RECP=RECPT@IBM.COM
DATA=&DATA1DD CID
DATA=&DATA2DD CID:IMAGE/JPEDG IMAGEID
DATA=&DATA3DD TXT SAMPLE.TXT
//DATA1DD   DD DISP=SHR, DSN=HLQ.CNTL (DATA1DD)
//DATA2DD   DD DISP=SHR, DSN=HLQ.CNTL (DATA2DD)
//DATA3DD   DD DISP=SHR, DSN=HLQ.CNTL (DATA3DD)
//MIMOUT    DD SYSOUT=(A, SMTP)
```

**Remarks:**

You have to specify REGION=0M in your JOB or EXEC CARD.



## 5.2.3 User Interface

### MIMOUT: DD

Description: Output DD containing MIME data which will be processed by SMTP application and sent to the recipients. E.g. SYSOUT = (A, < CSSMTP JOB NAME>).

### SNAPDUMP: DD

Description: Trace DD, in case requested, please do SYSOUT=\*, otherwise keep dummy to avoid diagnostics.

### CONFIGDD: DD

Description: Configuration DD, allowing users to customize the default EBCDIC and ASCII CCSID for input and output. If not provided, EBCDIC CCSID 1141 and ASCII CCSID 923 are used.

Format of configuration:

1. Lines start with '#' are token as Comments
2. DEFAULT\_EBCDIC\_CCSID is the keyword of default EBCDIC CCSID. The value after the '=' is token as the EBCDIC CCSID.
3. DEFAULT\_ASCII\_CCSID is the keyword of default EBCDIC CCSID. The value after the '=' is token as the EBCDIC CCSID.

An example of configuration:

```
# DEFAULT EBCDIC CCSID WORK AS INPUT
DEFAULT_EBCDIC_CCSID=1047
# DEFAULT ASCII CCSID WORK AS OUTPUT
DEFAULT_ASCII_CCSID=923
```

### PARMDD: DD

Description: You code all parameters in this statement. The syntax follows.

▶ parametername = parametervalue spaces ◀

<b>HOST</b>	Defines hostname, required but any value can be specified. Default: &LPAR. If &LPAR is specified, it must be starting parameter, and else it will be ignored.
<b>EHLO</b>	Default SMTP introduction clause, preferred for CSSMTP.
<b>HELO</b>	Defines SMTP HELO clause as HELO. (You need not specify those parameters for CSSMTP, EHLO is

	automatically added).
<b>TRAC</b>	Trace indicator, parameter can be omitted, setting default no trace. If present, must be second parameter. Possible values: 0 no trace (default) 8 maximal trace 9 extended trace (user data)
<b>SUBJ</b>	Subject parameter, taken as is till up to 109 characters if available. In case shorter than 109, automatically blanked up to 109th char. Includable: system variables &LPAR, &DATE, &TIME, &TIMO, &USER, &JOB, &JNUM. You can include those variables in HOST, MESSG, SUBJ and attachment name.
<b>MAXM</b>	Please specify an 8 byte number indicating the maximum message size being allowed. If the generated message has a larger length than this value, a return code of 008 will be issued. Sample: MAXM=20000000 for a limit of 20MB. If you specify a value less than 8 digits, it will be automatically padded with zeros at left.
<b>ORIG</b>	Sender of a mail message (maximum length 69), must precede DATA or MESSG, and follow SUBJ. Sample format: Either <a href="mailto:my.orig@domain.com">my.orig@domain.com</a> or "my name" <mail address>.
<b>RECP</b>	Recipient of a mail message (maximum length 69), must precede DATA or MESSG. For multiple recipients, specify multiple RECP statements. Referral to DD statement when specifying &<ddname>, and multiple referral allowed. Sample: RECP=&RECPNAME pointing to //RECPNAME DD * <a href="mailto:my.recipient@domain.com">my.recipient@domain.com</a> (multiple statements may follow)
<b>COPY</b>	CC recipient of a mail message (maximum length 69), must precede DATA or MESSG. For multiple recipients, specify multiple COPY statements. Referral to DD statement when specifying &<ddname>, and multiple referral allowed. Sample: COPY=&COPYNAME pointing to

```
//COPYNAME DD *
my.recipient.cc@domain.com
(multiple statements may follow)
```

**BCPY** BCC recipient of a mail message (maximum length 69), must precede DATA or MESH.

For multiple recipients, specify multiple BCPY statements. Referral to DD statement when specifying &<ddname>, and multiple referral allowed.

Sample:

```
BCPY=&BCPYNAME
pointing to
//BCPYNAME DD *
my.recipient.bcc@domain.com
(multiple statements may follow)
```

**RPLY** Reply-To recipient of a mail message (maximum length 69).

**SEND** Transmitter of a mail message (maximum length 69).

Must be specified, when author and transmitter of a message are not identical.

Name format:

Either

```
my.orig@domain.com
```

or

```
“my name” <mail address>.
```

**VARI** Changes value for Variable and address DD redirection prefix character. In some applications default value of & might conflict with application logic, e.g. OPC. You can specify the prefix as any 1-byte value (range 0x'00' to 0x'FF'). Scope is from next parameter statement. It is possible to provide this parameter several times in parameters in case you need this special character different in next parameters. It is up to user to choose proper character that will not conflict with other characters. No check is being done for user value of this parameter.

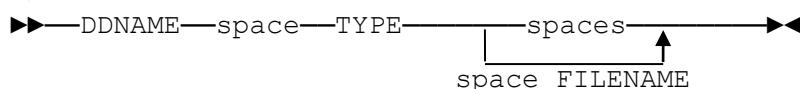
Default value: VARI=&.

Example:

```
ORIG=&ORIGNAME
VARI=~
RECP=~RECPNAME
```

**DATA** Describes data to be sent. For multiple bodies, code multiple DATA statements.

Syntax:



DDNAME DDNAME may be replaced by a DD statement variable, e.g. DATA=&SENDFILE. In this case the DD

statement SENDFILE must be presented in the JCL. Temporary data sets, instream data and concatenated data must be presented this way.

TYPE specifies type of the data to be sent. Possible values and corresponding meaning are shown in the following table:

Value	Meaning
TXT	EBCDIC input, ASCII output, CRLF added between lines, typically output from COBOL program or text file.
HTM	EBCDIC input, ASCII output, understands html and delivers inline html data or – if filename specified – html attachments.
BIN	Input is binary data, no data conversion is done, typically encrypted data, or pre-composed hex values.
BINxnnnn	nnnn stands for input code page, in case not present defaulted to cp1141.
x=U	Conversion to UTF-16 is performed upon input data.
x=8	Conversion to UTF-8 is performed upon input data.
x=A	Conversion to ASCII is performed upon input data.
	Example: BINU1141 to convert text from code page 1141 to UTF-16. BINA1153 to convert text from code page 1153 to ASCII cp923.
P7M	Input is binary data, no data conversion is done, encrypted data in enveloped format.
P7S	Input is binary data, no data conversion is done, signed data in enveloped format.
CID	HTML data, containing Location Identifiers being embedded as binary files from subsequent CID statements.

FILENAME if present, data will be interpreted as attachment and file name which will be used as name for this attachment. In case FILENAME is omitted, data is interpreted as inline text.

In case of CID: you must either leave the FILENAME empty (first CID value in the CID block, must be a HTML or text file specified in DDNAME), or starting from the second CID in the block followed by a multimedia type like 'image/jpeg', separated by ':', and then followed by a unique location identifier within the step, e.g. a file name like image1.jpg.

Includable: system variables &LPAR, &DATE, &TIME, &TIM0, &USER, &JOB, &JNUM: you can include those variables in HOST, MESS, SUBJ and attachment name.

Various formats for &DATE are available,

&DATx:

x=0: 'YYMMDD'

x=1: 'MM/DD/YY'

For x=other value, equals to value of &DATE:

'DD.MM.YYYY'.

Various formats for &TIME are available,

&TIMx:

x=E: 'HH:MM:SS'

x=0: 'HHMMSS'

(accepted as part of file name in attachments)

There are many types of multimedia files you can use, not only image/jpeg. Here is a link to a list of multimedia files:

<https://www.iana.org/assignments/media-types/media-types.xhtml>

<b>MESG</b>	<p>Describes data to be sent. Length is fixed 80 bytes, useful for short messages. Apostrophes not required.</p> <p>Includable: system variables &amp;LPAR, &amp;DATE, &amp;TIME, &amp;TIM0 &amp;USER, &amp;JOB, you can include those variables in HOST, MESS, SUBJ and attachment name.</p>
<b>CRLF</b>	<p>Sets new line sequence for ending one fetched record in case of text file. Scope of variable starts with next parameter. You can provide this parameter several times to change CRLF processing for different text blocks.</p> <p>Values:</p> <ul style="list-style-type: none"><li>0 Append nothing after text record.</li><li>1 Append hex '0D', this is default setting.</li><li>2 Append hex '0D0A'.</li></ul> <p>Example:</p> <p>CRLF=0 RTF applications, append nothing</p> <p>DATA=&amp;RTFDD TXT my_doc.rtf</p> <p>CRLF=2 append CRLF for windows notepad</p>

MESG=some text data line

MESG=another text data line

**ATXL**

Converts old X'7C' substitution characters in mail addresses to the @-sign. The code page must match the character of the @-sign. Italian and German characters '\$' are translated to '@' by ATXL=@\$. ATXL must be placed before ORIG/RECP. See also description in the appendix.

## 5.2.4 Comments in PARMDD

Parameter	Comment start column	Example
*		* Full line comment
HOST	After blank in value	HOST=&LPAR set always host=lparname where job runs
TRAC	7	TRAC=0 no trace
SUBJ	N/A	
ORIG	N/A*	* In case of DD redirection comment starts after specifying DD containing provided address record(s), DD length is always 8 characters, so comment starting column is 15.
RECP	N/A*	
COPY	N/A*	
BCPY	N/A*	
RPLY	N/A	
SEND	N/A	
DATA	N/A	
HELO	5	HELO this is for older SMTP clients
EHLO	5	EHLO CSSMTP recommended, default
MESG	N/A	
ATXL	8	ATXL=@\$ German at sign conversion
CRLF	7	CRLF=2 from now on append both CR and LF after record
MAXM	After blank	MAXM=900000 Maximum email size 9MB
VARI	VARI	VARI=\$ Use dollar as variable prefix

## 5.2.5 Return Codes

CUNMRCSM return codes:

- 0** Normal exit
- 4** Exit with warnings
- 8** Invalid parameter keyword or parameter length
- C** Internal error

## 5.2.6 Other Examples

### Remarks:

You have to specify REGION=0M in your JOB or EXEC CARD.

### Example 1

This example sends one short message and two datasets represented by DATASET1, DATASET2 to two recipients.

```
//STEP EXEC PGM=CUNMRCSM
//SNAPDUMP DD DUMMY --if needed, SYSOUT=*--
//PARMDD DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=Testmail-onemsg-2dsns-2recp
ORIG=SENDER@IBM.COM
RECP=RECPTER1@IBM.COM
RECP=RECPTER2@IBM.COM
MSG=Hello world, here is &USER on &LPAR at &DATE &TIME
DATA=&DATA1DD BINU1141 attmnt_1.txt
DATA=&DATA2DD BIN8 attmnt_2.txt
//DATA1DD DD DISP=SHR,DSN=HLQ.CNTL(DATA1DD)
//DATA2DD DD DISP=SHR,DSN=HLQ.CNTL(DATA2DD)
//MIMOUT DD SYSOUT=(A,SMTP)
```

### Example 2

This example sends one short message finished with an additional line feed. CRLF must be placed before MSG. And replied address is specified in RPLY:

```
//STEP EXEC PGM=CUNMRCSM
//SNAPDUMP DD DUMMY --if needed, SYSOUT=*--
//PARMDD DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ= Testmail-onemsg-CRLF
ORIG=SENDER@IBM.COM
RECP=RECPTER@IBM.COM
CRLF=2
MSG=Hello world, here is &USER on &LPAR at &DATE &TIME
//MIMOUT DD SYSOUT=(A,SMTP)
```

### Example 3

This example sends one short message finished with an additional line feed. Old German/Italian substitution character '\$' will be translated to '@'. ATXL must be placed before the addresses.

```
//STEP      EXEC PGM=CUNMRCSM
//SNAPDUMP   DD DUMMY      --if needed, SYSOUT=*--
//PARMDD     DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=Testmail-onemsg-ATXL
ATXL=$@
ORIG=SENDER$IBM.COM
RECP=RECPTER$IBM.COM
CRLF=2
MSG=Hello world, here is &USER on &LPAR at &DATE &TIME
//MIMOUT    DD SYSOUT=(A,SMTP)
```

### Example 4

This example sends data from a DD statement DDX referred to in the DATA parameter. The data will be sent as text attachment A&DATE.txt.

Optionally, both TXT and attachment name could be omitted, if normal text is requested only.

Recipients are listed in DD statement MYRECIPS.

```
//STEP      EXEC PGM=CUNMRCSM
//SNAPDUMP   DD DUMMY      --if needed, SYSOUT=*--
//PARMDD     DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=Testmail-DD
ORIG=SENDER@IBM.COM
RECP=&MYRECIPS
DATA=&DDX TXT A&DATE.txt
//DDX        DD DISP=SHR,DSN=HLQ.CNTL(DDX)
//MYRECIPS   DD DISP=SHR,DSN=HLQ.CNTL(RECIPS)
//MIMOUT     DD SYSOUT=(A,SMTP)
```

### Remarks:

ORIG has not been designed to point to a DD statement.



### Example 5

This example shows how to send multimedia data in an inline mail body using CUNMRCSM.

To achieve this goal, you have to compose a block consisting of a html input followed by binary files to be imbedded.

```
//STEP      EXEC PGM=CUNMRCSM
//SNAPDUMP DD DUMMY      --if needed, SYSOUT=*--
//PARMDD    DD *
HELO
HOST=SMTP.IBM.COM
TRAC=8
SUBJ=Testmail-HTML-multimedia
ORIG=SENDER@IBM.COM
RECP=RECPTER@IBM.COM
DATA=MY.HTML.DSN1 CID      the introducing HTML
DATA=MY.BIN.DSN2 CID:image/jpeg IMAGEID  embedded image
//MIMOUT    DD SYSOUT=(A,SMTP)
```

### Remarks:

The data must be in a format which HTML is able to handle. In this case it must be binary data. For the introducing HTML, it is EBCDIC text. For all other multimedia data, binary format is usually required. For CSV data, please convert your data to ASCII before using it.

There are many types of multimedia files you can use, not only image/jpeg. Here is a link to a list of multimedia files:

<https://www.iana.org/assignments/media-types/media-types.xhtml>.

## 5.3 Send Mail from COBOL Program

You can also put your data directly into a COBOL program (call CUNMCSMM module) to send user data to mail recipients.

Description and declaration in COBOL format:

```
*****
*
*      C O P Y - M E M B E R   F O R   C U N M C S M M   *
*
*****
01 XTSONL-RUECKPARM.
   *--- VALUE 0078 at entry to request HELO in SMTP protocol
05 XTSONL-RUECKPARM-LG          PIC 9(04)  VALUE 0077.
05 XTSONL-RETURN-CODE          PIC 9(03)  VALUE ZERO.
   *--- successful run
   88 XTSONL-RET-OK              VALUE ZERO.
   *--- wrong parameters
   88 XTSONL-RET-UEB            VALUE 008.
   *--- error in sub call
   88 XTSONL-RET-VERARB        VALUE 012.
   *--- error text
05 XTSONL-RETURN-TXT          PIC X(73)  VALUE SPACE.

01 XTSONL-ALLPARAM.
   *--- This parameter represents the server host name
   *--- containing the mail server functions.
03 XTSONL-HOSTPARAM.
   *--- Length of entire parameter
05 XTSONL-HOSTPARAM-LG        PIC 9(05)  VALUE 00149.
   *--- Argument for sub call
05 XTSONL-HOSTPARAM-TYP      PIC X(08)  VALUE 'HOSTPARAM'.
   *--- PORT, leave empty
05 XTSONL-HOSTPARAM-PORT     PIC 9(05)  VALUE ZERO.
   *--- Length of HOST parameter
05 XTSONL-HOSTPARAM-HOST-LG  PIC 9(03)  VALUE ZERO.
   *--- HOST parameter
05 XTSONL-HOSTPARAM-HOST     PIC X(128) VALUE SPACE.

   *--- This parameter contains the TRACE flag
03 XTSONL-UEBPARMF.
   *--- Length of entire parameter
```

```

05 XTSONL-UEBPARMF-LG          PIC 9(05)  VALUE 00014.
*--- Argument for sub call
05 XTSONL-UEBPARMF-TYP        PIC X(08)  VALUE 'UEBPARMF '.
*--- Trace level 0 or 8
05 XTSONL-UEBPARMF-TRACE-LVL  PIC 9(01)  VALUE ZERO.
    88 XTSONL-KEIN-TRACE          VALUE ZERO.
    88 XTSONL-MAX-TRACE          VALUE 8.
    88 XTSONL-EXT-TRACE          VALUE 9.

*--- This parameter group contains a number of fixed and
*--- variable times callable parameters, specifying the
*--- appropriate arguments for subject, originator and
*--- recipient(s)
*--- UEBPARMV reserved max length 6800          (variable)
*--- Header incl. entire length: 13          (variable)
*--- Subj length incl. header and length 120   (fixed)
*--- ORIG length incl. header and length 80    (fixed)
*--- RECP length incl. header and length 80
*--- (max 80 recps/replyto, each fixed Length 80 = max 6400)
03 XTSONL-UEBPARMV.
    *--- Length of entire parameter set used (max 6800)
    05 XTSONL-UEBPARMV-LG          PIC 9(05)  VALUE ZERO.
    *--- Argument for sub call
    05 XTSONL-UEBPARMV-TYP        PIC X(08)  VALUE 'UEBPARMV '.
    *--- Subject parameter fixed length 120
    05 XTSONL-UEBPARMV-TYP-S
        07 XTSONL-SUBJECT          PIC X(08)  VALUE 'SUBJECT '.
        07 XTSONL-SUBJECT-LG       PIC 9(03)  VALUE ZERO.
        07 XTSONL-SUBJECT-FIELD    PIC X(109) VALUE SPACE.
    *--- Originator parameter fixed length 80
    05 XTSONL-UEBPARMV-TYP-O
        07 XTSONL-ORIG             PIC X(08)  VALUE 'FROM '.
        07 XTSONL-ORIG-LG         PIC 9(03)  VALUE ZERO.
        07 XTSONL-ORIG-FIELD      PIC X(69)  VALUE SPACE.
    *--- Recipient parameter each fixed length 80,
    *--- max 80 recipients when recipient list
    05 XTSONL-UEBPARMV-TYP-R
        07 XTSONL-RECP             PIC X(08)  VALUE SPACE.
        88 XTSONL-RECP-TO         VALUE 'TO '.
        88 XTSONL-RECP-CC         VALUE 'CC '.
        88 XTSONL-RECP-BCC        VALUE 'BCC '.
        88 XTSONL-RECP-REPLY      VALUE 'REPLY '.
        88 XTSONL-RECP-SENDER     VALUE 'SENDER '.
        07 XTSONL-RECP-LG         PIC 9(03)  VALUE ZERO.

```

07 XTSONL-RECP-FIELD PIC X(69) VALUE SPACE.

\*--- This parameter is for mail data.

03 XTSONL-DATA.

\*--- Length of entire DATA parameter

\*--- It is not evaluated. User can put there any value

\*--- for documentation, but it must be 8 digits.

\*--- If not used, specify 8 zeros.

05 XTSONL-DATA-LG PIC 9(08).

\*--- Argument for sub call

05 XTSONL-DATA-TYP PIC X(08) VALUE 'DATA '.

\*--- Mark (E), if this is last data body part. Default

\*--- Mark (M), if there are additional body parts.

05 XTSONL-DATA-FLAG PIC X(01) VALUE SPACE.

88 XTSONL-DATA-END VALUE 'E'.

88 XTSONL-DATA-MORE VALUE 'M'.

\*--- Mark if attachment.

05 XTSONL-DATA-ATT-FLAG PIC X(01) VALUE SPACE.

88 XTSONL-DATA-ATTACHMENT VALUE 'A'.

88 XTSONL-DATA-TEXT VALUE 'T'.

\*--- Attachment name

05 XTSONL-DATA-ATTNAME PIC X(64) VALUE SPACE.

\*--- For embedded files MIME type only:

07 XTSONL-DATA-ATTNAME-CONTENT-TYPE PIC X(24).

07 XTSONL-DATA-ATTNAME-LOCATION PIC X(14).

07 XTSONL-DATA-MIME-FILLER PIC X(26) VALUE SPACE.

\*--- MIME type

\*--- TXT for Text data, default.

\*--- HTM for HTML data.

\*--- BINU for Text data.

\*--- BIN8 for Text data.

\*--- BIN for Binary data.

\*--- BINA for Text data.

\*--- P7M for Binary encrypted data in enveloped format.

\*--- P7S for Binary signed data in enveloped format.

\*--- CIDs for embedded HTML.

\*--- All the CIDS shall be consecutive.

\*--- The first CID consisting of HTML data, and

\*--- consecutive ones consisting of Binary data.

\*--- No character conversion will be done for Binary data.

\*--- Both HTML and Text data will be converted accordingly.

\*--- HTM/TXT is converted from Default EBCDIC to default ASCII.

\*--- BINU is converted from MIME-TYP-CCSID to UTF-16.

\*--- BIN8 is converted from MIME-TYP-CCSID to UTF-8.

```

*--- BINA is converted from MIME-TYP-CCSID to default ASCII.
05 XTSONL-DATA-MIME-TYP PIC X(64) VALUE SPACE.
    07 XTSONL-DATA-MIME-TYP-MIME PIC X(03) .
    07 XTSONL-DATA-MIME-TYP-UNI PIC X.
        88 XTSONL-DATA-MIME-TYP-UNIUTF-16 VALUE 'U'.
        88 XTSONL-DATA-MIME-TYP-UNIUTF-8 VALUE '8'.
        88 XTSONL-DATA-MIME-TYP-ASCII-ISO8859 VALUE 'A'.
    07 XTSONL-DATA-MIME-TYP-CCSID PIC X(04) VALUE 1141.
    07 XTSONL-DATA-MIME-FILLER PIC X(56) VALUE SPACE.
*--- Length of data (theoretical max 99999999)
*--- Minimum size in binary files 40 bytes,
*--- else unpredictable results
05 XTSONL-DATA-STRING-LG PIC 9(08) VALUE ZERO.
*--- Data contents
05 XTSONL-DATA-STRING PIC X(max. 99999999) VALUE SPACE.

```

### Detail Description:

#### Reg. TRACE-LVL:

Coding trace value '8' will show the protocol process in detail.

Coding trace value '9' will show user data in addition to that and some other CUNMCSMM program breakpoint data important for error diagnosis.

#### Reg. RECP/REPLY/SENDER/CC/BCC and ORIG:

Please specify real length so no blanks are generated.

Even if the declaration of the various address fields are 69 characters, you should specify the real length in the leading length field (e.g. XTSONL-RECP-LG ).

The SUBJ value is taken as it is, you can specify anything in the 109 characters, even blanks.

Its length is a comment only and not evaluated. It is recommended to specify a value of 109.

#### Reg. DATA:

You can code up to 100 data blocks.

XTSONL-DATA-FLAG must be 'E' in last block of chain.

DATA blocks following an 'E' marked block, are ignored.

Specifying 'M' in last DATA block causes RC8 ('no DATA buffer'), but may cause unpredictable results.

The multiple body parts may be mixed either as binary or text, inline or attachment. Binary data should be specified as an attachment, because the capability of the receiving mail server may not support binary data in an inline format.

#### Reg. UNICODE:

Specify 'BIN' as XTSONL-DATA-MIME-TYP-MIME,

use 'U' as XTSONL-DATA-MIME-TYP-UNI for UTF-16,

use '8' as XTSONL-DATA-MIME-TYP-UNI for UTF-8.

### Coding the user program:

- The entry point being called is 'CUNMCSMX'.

- Please take a valid 'FROM' address.
- The '@' sign may require a substitution character like '\$' in some national code pages. Please refer to the CODEPAGE value in CSSMTP.
- To indicate a new line within the data part,
  - code X'0D', for XTSONL-DATA-MIME-TYP=TXT,
  - code X'0D15' for XTSONL-DATA-MIME-TYP=BIN8 and XTSONL-DATA-MIME-TYP=BINU.

(in batch program CUNMRCSH this is done automatically)

## Appendix 1

The logic behind the @-sign has a historical aspect.

Mail server software usually had CP1047 as home code page. This code page is called 'Latin1 Opens Systems', it is nearly identical with US and UK English code pages. The @-sign is located at code point X'7C'.

Using other national code pages, users had to use the character, being located on X'7C', as substitution character for the @-sign.

Here you have a table of some code pages and the affected character:

German	1141	§
Swedish	1143	Ö
Icelandic	1149	Ð
French	1147	à
Norwegian	1142	Ø

Others have the identical sign as the code pages mentioned above, like Italian = German '§'.

Some code pages have the @-sign as default '@': Spanish, Latin2 (Czech.).

If you use those substitution characters from X'7C' above, and keep code page 1047 as code page for CSSMTP, the work flow will be flawless.

If you use the native @ character from your national code page, and set up your national code page in CSSMTP, like 1141 for Germany, it will work.

However, if you use the old X'7C' substitution characters and set up your national code page in CSSMTP not having the @-sign at this location, it does not work.

RUNCSMIM converts X'7C' to the appropriate character in the national code page being used in CSSMTP, when you specify the parameter ATXL:

German	1141	§	-> X'B5'	ATXL=§@
Swedish	1143	Ö	-> X'EC'	ATXL=Ö@
Icelandic	1149	Ð	-> X'AC'	ATXL=Ð@
French	1147	à	-> X'44'	ATXL=à@
Norwegian/Danish	1142	Ø	-> X'80'	ATXL=Ø@

Using parameter ATXL in RUNCSMIM has no compromising effects, it affects the translation of the character in the mailing addresses only, not in the data itself.

Observe: ATXL must be specified **before** ORIG/RECP parameters in PARMDD.

## Appendix 2

Code page translation using Unicode Systems Services:

(Code page specified in CSSMTP translate statement will affect mail addresses only.)

