```
**FREE
ctl-opt option (*srcstmt : *nodebugio : *nounref);
ctl-opt debug (*input);
ctl-opt dftactgrp (*no);
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
//  Program : MlrSndEml
//  Author  : Mike Larsen
//  Date Written: 06/14/2020
//  Purpose : This program will send an Html formatted email to a
//            recipient as well as an attachment.
//
//  Notes   : cannot exceed 400 characters in the current version of the
//            SNDSMTPEMM command
//
//=========================================================================*
//   Date    Programmer  Description                                       *
//-------------------------------------------------------------------------*
// 06/14/20  M.Larsen    Original code.                                    *
//                                                                         *
//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *

// - - - -
// Workfields

dcl-s recipient      char(50);
dcl-s subject        char(150);
dcl-s body           char(50);
dcl-s CmdStr         char(3000);
dcl-s Quote          char(1) inz('''');
dcl-s htmlBreak      char(4) inz('<br>');
dcl-s vHtmlSignature varchar(400);
dcl-s fileType       char(5);

// - - - -
// variables to hold email values

dcl-s emailClosing  char(50);
dcl-s emailName     char(50);
dcl-s emailTitle    char(50);
dcl-s emailCompany  char(50);
dcl-s emailAddress1 char(100);
dcl-s emailAddress2 char(100);

// - - - -
// for the Html signature

dcl-s signaturePath char(50);

// - - - -
// for the jpeg

dcl-s jpegPath     char(50);
dcl-s jpegFileName char(50);

dcl-s File_In Sqltype(Clob_file);
```

```
// - - - - - - -

// The SQLTYPE(CLOB_FILE) definition will be converted by the compiler
// into the following data structure:

// DCL-DS FILE_IN;
//   FILE_IN_NL UNS(10);               // length of the value from File_In_NAME
//   FILE_IN_DL UNS(10);               // data length
//   FILE_IN_FO UNS(10);               // type of file open
//   FILE_IN_NAME CHAR(255) CCSID(*JOBRUNMIX); // IFS file I'm uploading
// END-DS FILE_IN;

// - - - -
// for the scanning and replacing of the Html signature variables

dcl-s closing   char(20) inz('closing');
dcl-s name      char(20) inz('name');
dcl-s title     char(20) inz('title');
dcl-s company   char(20) inz('company');
dcl-s address1  char(20) inz('address1');
dcl-s address2  char(20) inz('address2');

// - - - -
// Run CL Command

dcl-pr Run     ExtPgm('QCMDEXC');
       CmdStr  Char(3000)   Options(*VarSize);
       CmdLen  Packed(15:5) Const;
       CmdDbcs Char(2)      Const Options(*Nopass);
End-pr;

//-------------------------------------------------------

populateEmailVariables();
getHtmlSignature();
replaceSignatureVariables();
sendEmail();

*Inlr = *On;
Return;

//-------------------------------------------------------
// populateEmailVariables subprocedure
//-------------------------------------------------------

dcl-proc populateEmailVariables;

  // in a production program, these would be soft-coded

  emailClosing  = 'Sincerely, ';
  emailName     = 'Mike Larsen';
  emailTitle    = 'Project Manager & Senior developer';
  emailCompany  = 'Central Park Data Systems';
  emailAddress1 = '555 Main Street';
```

```
  emailAddress2 = 'Long Island, NY 11101';
  body          = 'Look at this awesome photo of Lexie!';

end-proc populateEmailVariables;

//-------------------------------------------------------
// getHtmlSignature subprocedure
//-------------------------------------------------------

dcl-proc getHtmlSignature;

  signaturePath = '/home/MLARSEN/Email_sig.txt';

  // set up output parameters.  we'll take a document from the IFS and
  // and load it into a Clob variable.

  File_In_fo   = Sqfrd;
  File_In_name = %trim(signaturePath);
  File_In_nl   = %Len(%trimR(File_In_name));

  // Populate the 'vHtmlSignature' variable from the Ifs Clob variable
  // 'File_In'.

  Exec sql
    Select :File_In
     Into :vHtmlSignature
     from sysibm.sysdummy1;

end-proc getHtmlSignature;

//-------------------------------------------------------
// replaceSignatureVariables subprocedure
//-------------------------------------------------------

dcl-proc replaceSignatureVariables;

  vHtmlSignature = %ScanRpl(%trim(closing) : %trim(emailClosing)
                            : vHtmlSignature);

  vHtmlSignature = %ScanRpl(%trim(name) : %trim(emailName)
                            : vHtmlSignature);

  vHtmlSignature = %ScanRpl(%trim(title) : %trim(emailTitle)
                            : vHtmlSignature);

  vHtmlSignature = %ScanRpl(%trim(company) : %trim(emailCompany)
                            : vHtmlSignature);

  vHtmlSignature = %ScanRpl(%trim(address1) : %trim(emailAddress1)
                            : vHtmlSignature);

  vHtmlSignature = %ScanRpl(%trim(address2) : %trim(emailAddress2)
                            : vHtmlSignature);

  vHtmlSignature = %trim(body) + %trim(htmlBreak) + %trim(vHtmlSignature);
```

```
end-proc replaceSignatureVariables;

//-------------------------------------------------------
// sendEmail subprocedure
//-------------------------------------------------------

dcl-proc sendEmail;

  jpegPath     = '/home/MLARSEN';
  jpegFileName = 'Lexie';
  fileType     = '.jpg';
  subject      = 'This email was sent from IBM i';
  recipient    = 'mlarsen@centralparkdata.com';

  CmdStr = 'SNDSMTPEMM RCP(' +
            Quote + %trim(recipient) +
            Quote + ') ' +
           'SUBJECT(' + Quote + %trim(subject) + Quote + ') ' +
           'NOTE(' + Quote + %trim(vHtmlSignature) + Quote + ')' +
           ' ATTACH((' + Quote + %trim(jpegPath) + '/' +
          %trim(jpegFileName) + %trim(fileType) + Quote +
           ' *OCTET *BIN))' +
           ' CONTENT(*HTML)';

  Callp Run(Cmdstr:%Size(CmdStr));

end-proc sendEmail;

//- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```