

```

/*****
/* Program name . . . . . : CVTQRY2IFS */
/* Author . . . . . : Raymond G. Johnson */
/* Date created . . . . . : 08/12/2007 */
/*
/*
/* Program information: Convert Query output to CSV file in the IFS.
/*
/*
/*
/*****
/* Jim Sloan Error recovery procedure.
/*****
/* Change log:
/*****
/* Date Initials Change Description
/*****
/* xx/xx/200x XXX
/*
/*
/*
/*****

PGM PARM(&QRYNAM &QRYLIB)
DCL VAR(&ERRORSW) TYPE(*LGL) /*Std Error */
DCL VAR(&MSGID) TYPE(*CHAR) LEN(7) /*Std Error */
DCL VAR(&MSG) TYPE(*CHAR) LEN(512) /*Std Error */
DCL VAR(&MSGDTA) TYPE(*CHAR) LEN(512) /*Std Error */
DCL VAR(&MSGF) TYPE(*CHAR) LEN(10) /*Std Error */
DCL VAR(&MSGFLIB) TYPE(*CHAR) LEN(10) /*Std Error */
DCL VAR(&KEYVAR) TYPE(*CHAR) LEN(4) /*Std Error */
DCL VAR(&KEYVAR2) TYPE(*CHAR) LEN(4) /*Std Error */
DCL VAR(&RTNTYPE) TYPE(*CHAR) LEN(2) /*Std Error */

DCL VAR(&QRYNAM) TYPE(*CHAR) LEN(10)
DCL VAR(&IFSNAM) TYPE(*CHAR) LEN(80)
DCL VAR(&QRYLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&IFSDIR) TYPE(*CHAR) LEN(10)

/*****
/* Global Monitor Message - Standard error exit
/*****

MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(STDERR1))

/*****
/* Make sure that the query actually exists.
/*
/* If not tell the user and exit.
/*****

CHKOBJ OBJ(&QRYLIB/&QRYNAM) OBJTYPE(*QRYDFN)
MONMSG MSGID(CPF9801 CPF9810) EXEC(DO
  SNDPGMMSG MSG('The query ' *BCAT &QRYNAM *BCAT 'in +
    library ' *BCAT &QRYLIB *BCAT 'was not +
    found. Please try again.')
  GOTO RETURN
ENDDO

/*****
/* Identify the target directory for XLS file.
/*
/* These directories must exist.
/*
/* These directories are not checked for existence.
/*****

SELECT
  WHEN COND(&QRYLIB = 'QRYLIB1') THEN(CHGVAR +
    VAR(&IFSDIR) VALUE('/xls/ap'))
  WHEN COND(&QRYLIB = 'QRYLIB2') THEN(CHGVAR +
    VAR(&IFSDIR) VALUE('/xls/ar'))
  WHEN COND(&QRYLIB = 'QRYLIB3') THEN(CHGVAR +
    VAR(&IFSDIR) VALUE('/xls/gl'))
ENDSELECT

```

```

/*****/
/*      Delete the old query if it exists in QTEMP.          */
/*      This will make the program recursive even if the user */
/*      changes the original query on the fly.                */
/*****/

      DLTQRY      QRY (QTEMP/&QRYNAM)
      MONMSG      MSGID (CPF2105)

/*****/
/*      Copy the query to QTEMP library.                      */
/*****/

      CRTDUPOBJ  OBJ(&QRYNAM) FROMLIB(&QRYLIB) OBJTYPE(*QRYDFN) +
                TOLIB(QTEMP)

/*****/
/*      Run the query with output to a temporary file.        */
/*****/

      RUNQRY     QRY (QTEMP/&QRYNAM) OUTTYPE(*OUTFILE) PRITDFN(*NO) +
                OUTFILE(QTEMP/&QRYNAM *FIRST *RPLFILE) AUT(*USE)

/*****/
/*      Create the name for the file in the IFS directory.     */
/*****/

      CHGVAR     VAR(&IFSNAM) VALUE (&IFSDIR *TCAT '/' *TCAT &QRYNAM +
                *TCAT '.csv')

/*****/
/*      Convert the resulting physical file into a CSV type file. */
/*****/

      CPYTOIMPF  FROMFILE (QTEMP/&QRYNAM) TOSTMF (&IFSNAM) +
                MBROPT(*REPLACE) STMFCDPAG(*PCASCII) +
                RCDDL(*CRLF)

/*****/
/*      Change the authority since the newly created file will most */
/*      likely NOT have the authority that you expect.            */
/*****/

      CHGAUT     OBJ(&ifsnam) USER(*PUBLIC) DTAAUT(*RWX) +
                OBJAUT(*ALL)

/*****/
/*      Clean up the QTEMP library.                              */
/*      It's always nice to clean up after yourself.            */
/*****/

      DLTQRY     QRY (QTEMP/&QRYNAM)

RETURN:      RETURN /* Normal end of program. */
/*****/
/*      END OF PROGRAM / ERROR EXIT                             */
/*****/

STDERR1:
      IF        COND (&ERRORSW) THEN (SNDPGMSG MSGID (CPF9999) +
                MSGF (QCPFMSG) MSGTYPE (*ESCAPE))
      CHGVAR    VAR (&ERRORSW) VALUE ('1') /* Set to FAIL on error */
      RCVMSG    MSGTYPE (*EXCP) RMV (*NO) KEYVAR (&KEYVAR)

STDERR2:
      RCVMSG    MSGTYPE (*PRV) MSGKEY (&KEYVAR) RMV (*NO) +
                KEYVAR (&KEYVAR2) MSG (&MSG) MSGDTA (&MSGDTA) +
                MSGID (&MSGID) RTNTYPE (&RTNTYPE) MSGF (&MSGF) +
                SNDMSGFLIB (&MSGFLIB)
      IF        COND (&RTNTYPE *NE '02') THEN (GOTO STDERR3)
      IF        COND (&MSGID *NE ' ') THEN (SNDPGMSG MSGID (&MSGID) +
                MSGF (&MSGFLIB/&MSGF) MSGDTA (&MSGDTA) +

```

```
                MSGTYPE (*DIAG)
IF              COND (&MSGID *EQ ' ') THEN (SNDPGMMMSG MSG (&MSG) +
                MSGTYPE (*DIAG)
RMVMSG         MSGKEY (&KEYVAR2)
```

```
STDERR3:      RCVMSG   MSGKEY (&KEYVAR) MSGDTA (&MSGDTA) MSGID (&MSGID) +
                MSGF (&MSGF) SNDMSGFLIB (&MSGFLIB)
                SNDPGMMMSG MSGID (&MSGID) MSGF (&MSGFLIB/&MSGF) MSGDTA (&MSGDTA) +
                MSGTYPE (*ESCAPE)
```

```
END:          ENDPGM
```