
Subject: Re: McIDAS anyone?

Posted by [Justin Baker](#) on Tue, 18 Mar 1997 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

This is a multi-part message in MIME format.

-----167E2781446B

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

Howard,

Kelly Dean sent me this routine in response to a similar request some time ago (I hope that's OK Kelly !).

Regards,
Justin.

Regional Computing Development Section
Bureau of Meteorology
email: j.baker@bom.gov.au

-----167E2781446B

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

Content-Disposition: inline; filename="mcimage.pro"

FUNCTION READ_McDATA, unit, nx, ny, ipzero, prefix, bytes, McIMAGE

ON_IOERROR, BAD

IF (BYTES EQ 1) THEN BEGIN

 McIMAGE = BYTARR(nx+prefix/2,ny)

 POINT_LUN, unit, ipzero

 READU, unit, McIMAGE

 McIMAGE = ROTATE(TEMPORARY(McIMAGE),7)

ENDIF ELSE BEGIN

 McIMAGE = INTARR(nx+prefix/2,ny)

 POINT_LUN, unit, ipzero

 READU, unit, McIMAGE

 McIMAGE = \$

 ROTATE(TEMPORARY(ISHFT(TEMPORARY(McIMAGE(prefix/2:nx+prefix/ 2-1,*)),-5)),7)

ENDELSE

;

; Return with status.

;

status = 1

GOTO, DONE

BAD: PRINT, !ERR_STRING & status = 0

DONE: RETURN, status

END ; READ_McDATA

FUNCTION READ_McHEAD, unit, nx, ny, ipzero, prefix, bytes, ichanl

;
; PURPOSE:
; FUNCTION READ_McHEAD will read the PC McIDAS formatted image files
; developed at CIRA.

;
; The mcidas data is organized as:
; HEADER first byte = 0
; CALIBRATION first byte =
; NAVIGATION first byte = pri_key_calib
; DATA: prefix, pixels first byte = pri_key+nav

;
; CALLING SEQUENCE:
; To use this open the file [position at byte 0 for multiple reads]

;
; INPUTS:
; unit: Unit number of disk file

;
; OUTPUTS:
;
; ipzero : Starting byte of image [including prefix]
; nx : Width of image in pixels
; ny : Height of image in lines
; prefix :
; bytes : Bytes per pixel [1 or 2]
; ichanl : Image channel number [1,2,3,4,5]

;
; Function READ_PSE returns the status for the read.
; status = 1 TRUE
; status = 0 FALSE

;
; EXAMPLE:
;
; status = READ_McREAD(unit)

;
; RESTRICTIONS:
; This is our first attempt to read GOES-NEXT GVAR data formatted in PC
; McIDAS developed internally to read GOES-NEXT imagery collected here at
; CIRA. We do not guarantee that this is the standard McIDAS format. This
; format or information within this format is subject to change as
; further check out is going on with GOES-8.

;
; MODIFICATION HISTORY:
; Written by: Garrett Campbell and Kelly Dean, June 1994

```

; Colorado State University/CIRA
;
;
;-
ON_IOERROR, BAD
;
; Define McIDAS header structure
;
McHEADstr = { McHEAD, $
    area_status:0L, $
    version_num:0L, $
    sat_id_num:0L, $
    img_date:0L, $
    img_time:0L, $
    north_bound:0L, $
    west_vis_pixel:0L, $
    z_coor:0L, $
    num_line:0L, $
    num_elem:0L, $
    bytes_per_pixel:0L, $
    line_res:0L, $
    elem_res:0L, $
    num_chan:0L, $
    num_byte_in_prefix:0L, $
    proj_num:0L, $
    creation_date:0L, $
    creation_time:0L, $
    sndr_filter_map:0L, $
    img_id_num:0L, $
    id:lonarr(4), $
    comment:string(' ',format='(a32)'), $
    pri_key_calib:0L, $ ; end of calibration block
    pri_key_nav:0L, $ ; end of navigation block
    sec_key_nav:0L, $
    val_code:0L, $
    pdl:lonarr(8), $
    band8:0L, $
    act_img_date:0L, $
    act_img_time:0L, $
    act_start_scan:0L, $
    len_prefix_doc:0L, $
    len_prefix_calib:0L, $
    len_prefix_lev:0L, $
    src_type:' ', $
    calib_type:' ', $
    avg_or_sample:0L, $
    poes_signal:0L, $
    poes_up_down:0L, $
    orig_src_type:' ', $

```

```

reserved:lonarr(7)}

ReadU, unit, McHEADstr

ipzero = McHEADstr.pri_key_nav ; Starting byte of image [including prefix]
nx      = McHEADstr.num_elem  ; Width of image in pixels
ny      = McHEADstr.num_line  ; Height of image in lines
prefix  = McHEADstr.num_byte_in_prefix
bytes   = McHEADstr.bytes_per_pixel ; Bytes per pixel [1 or 2]
ichanl  = McHEADstr.img_id_num ; Image channel number [1,2,3,4,5]

help, /structure, McHEADstr
;print,nx,ny,ipzero,prefix,bytes,ichanl

;
;
; Return with status.
;
status = 1
GOTO, DONE
BAD: PRINT, !ERR_STRING & status = 0
DONE: RETURN, status

END ; READ_McHEAD

PRO MCTEST

;filename = 'f:\temp\u9414113.c04'
;filename = 'f:\temp\u9414113.c01'
;filename = 'd16:[jungle.erbe]TEST.C04'
filename = pickfile(/read, filter = 'AREA*')

OpenR, unit, filename, /GET_LUN

PRINT, 'Reading McIDAS image....'

IF ( READ_McHEAD( unit, nx, ny, ipzero, prefix, bytes, ichanl) ) THEN BEGIN
PRINT, ' File name >', filename
PRINT, ' NX      >', nx
PRINT, ' NY      >', ny
PRINT, ' IPzero  >', ipzero
    PRINT, ' Prefix >', prefix
    PRINT, ' Bytes  >', bytes
PRINT, ' IChanl  >', ichanl
ENDIF ELSE BEGIN
PRINT, 'This is not a McIDAS file >', filename
ENDELSE

IF ( READ_McDATA( unit, nx, ny, ipzero, prefix, bytes, McIMAGE) ) THEN BEGIN

```

```
PRINT, 'I read McIDAS file >', filename
; small = EXTRAC(McIMAGE(0), prefix, ny, nx, ny)
small = CONGRID(McIMAGE, nx/2, ny/2 )
    WINDOW, 0, xsize = nx/2, ysize = ny/2, title = filename
    TVscl, small
ENDIF ELSE BEGIN
PRINT, 'This is not a McIDAS file >', filename
ENDELSE
```

END

-----167E2781446B--
