

More than a Person Might Ever Want to Know about the WCS

The World Coordinate System is a wonderful and wacky thing. When one takes a Mosaic image, a default WCS is written into your header, which includes basic scale, rotation, and higher order terms.

The zeropoint of this system however is defined by a combination of these terms AND the values of TELRA and TELDEC in the header.

Copying wcs from one image to another

To **copy the wcs from one image to another** (after you have screwed it up) use wscopy. This can be invoked as follows:

msccmd "wscopy \$input \$in2 verbose+"

The first list ("input") will be the image you want to modify; the second list ("in2") is the reference image.

What do the higher-order terms actually mean?

Lindsey Davis was kind enough to spell this out; her note is reproduced verbatim here:

The TNX projection is computed as follows:

1. Compute the first order standard coordinates xi and eta from the linear part of the solution stored in CRPIX and the CD matrix.

$$\begin{aligned} \text{xi} &= \text{CD1_1} * (\text{x} - \text{CRPIX1}) + \text{CD1_2} * (\text{y} - \text{CRPIX2}) \\ \text{eta} &= \text{CD2_1} * (\text{x} - \text{CRPIX1}) + \text{CD2_2} * (\text{y} - \text{CRPIX2}) \end{aligned}$$

2. Apply the non-linear part of the solution if any (the part stored in the lngcor and latcor strings in the WAT keywords).

$$\begin{aligned} \text{xi}' &= \text{xi} + \text{lngcor}(\text{xi}, \text{eta}) \\ \text{eta}' &= \text{eta} + \text{latcor}(\text{xi}, \text{eta}) \end{aligned}$$

3. Apply the standard tangent plane projection to xi' and eta' using the CRVAL values as the tangent point.

The main difficulty is interpreting the numbers in lngcor and latcor. These numbers represent a surface fit for each coordinate with the entries as follows.

1. the first number is the surface type encoded as a real (1=chebyshev, 2=legendre, 3=polynomial). Your header has 3 in this spot which is the easiest surface polynomial to deal with.

2. the next two numbers represent the order of the fit in x and y encoded as reals. In your case both are 4 which means that cubic polynomials in xi and eta were fit.

3. the next number defines the type of cross-terms encoded as a real (0=no cross-terms, 1=full cross-terms, 2=half-cross terms). You have half cross-terms which for a cubic polynomial of order 4 and 4 means you will have 10 terms in each surface.

4. the next 4 numbers describe the region of validity of the fits in xi and eta space, e.g. xmin, xmax, etamin, etamax.

5. The next 10 terms are the coefficients of the polynomial terms, i.e. 1, xi, xi**2, xi**3, eta, xi*eta, xi**2*eta, eta**2, eta**2*xi, and eta**3, respectively.

The units of xi, eta are degrees. The CRVALS are in degrees and the CD matrix elements are in degrees / pixel.

Since I suspect your header is fairly standard I hope the above is enough info to get you started decoding. You will have to do more work if:

1. the surface fit was legendre or chebyshev. In this case you will need to normalize the xi and eta coordinates going into lngcor and latcor to the range -1.0 to 1.0 using ximin, ximax, etamin, etamax and of course compute the appropriate polynomial terms.

2. if your cross-terms options is not 2 then the number of terms will be different, i.e. 7 if cross-terms = 0 and 16 if cross-terms=2.

If your data doesn't match the header below let me know and I will send more information.

Sample: header

```
WCSASTRM= 'ct4m.19990714T012701 (USNO-K V) by F. Valdes 1999-08-02' / WCS Source
WCSDIM    =                2 / WCS dimensionality
CTYPE1    = 'RA---TNX'       / Coordinate type
CTYPE2    = 'DEC--TNX'       / Coordinate type
CRVAL1    =   310.08145293602507 / Coordinate reference value
CRVAL2    =   20.663666538998399 / Coordinate reference value
CRPIX1    =    4268.3258 / Coordinate reference pixel
CRPIX2    =    2256.2481 / Coordinate reference pixel
CD1_1     =   -6.8295807e-08 / Coordinate matrix
CD2_1     =    7.3313414e-05 / Coordinate matrix
CD1_2     =    7.374228e-05 / Coordinate matrix
CD2_2     =   -1.1927219e-06 / Coordinate matrix
WAT0_001= 'system=image'     / Coordinate system
WAT1_001= 'wtype=tnx axtype=ra lngcor = "3. 4. 4. 2. -0.3171856965643079 -0.015
',
WAT1_002= '0652479325533 -0.3126038394350166 -0.1511955040928311 0.002318100364
',
WAT1_003= '838772 0.01749134520424022 -0.01082784423020123 -0.1387962673564234
',
WAT1_004= '-4.307309762939804E-4 0.009069288008295441 0.002875265278754504 -0.0
',
WAT1_005= '4487658756007625 -0.1058043162287004 -0.0686214765375767 "
',
WAT2_001= 'wtype=tnx axtype=dec latcor = "3. 4. 4. 2. -0.3171856965643079 -0.01
',
WAT2_002= '50652479325533 -0.3126038394350166 -0.1511955040928311 0.00553481957
',
WAT2_003= '8784082 0.01258790793029932 0.01016780085575339 0.01541083298696018
',
WAT2_004= '0.03531979587941362 0.0150096457430599 -0.1086479352595234 0.0399806
',
WAT2_005= '086902122 0.02341002785565408 -0.07773808393244387 "
',
```

Below is some more information about the exact form of the polynomials generated by ccmmap / geomap and stored in the image header, which may be useful to you if you try to deal with the chebyshev and legendre forms ... Note the items under surface 2 are exactly analogous to the items

in the lngcor and latcor attributes in the WAT keywords. The items under surfacel are translated into the appropriate CRVAL, CRPIX, and CD matrix values.

#####

1. For simple linear geometric transformations you will see the following two entries in the fit record. Surfacel describes the linear portion of the fit; surface2 describes the residual distortion map which is always 0 for linear fits.

```

surfacel      11
  surface(xfit) surface(yfit)      (surface type 1=cheb, 2=leg, 3=poly)
  xxorder(xfit) yxorder(yfit)      (always 2)
  xyorder(xfit) yyorder(yfit)      (always 2)
  xxterms(xfit) yxterms(yfit)      (always 0)
  xmin(xfit)    xmin(yfit)          (geomap input or data)
  xmax(xfit)    xmax(yfit)          (geomap input or data)
  ymin(xfit)    ymin(yfit)          (geomap input or data)
  ymax(xfit)    ymax(yfit)          (geomap input or data)
  a             d
  b             e
  c             f
surface2      0

```

The above record describes the following linear surfaces.

```

xfit = a + b * x + c * y  (polynomial)
yfit = d + e * x + f * y

xfit = a + b * xnorm + c * ynorm  (chebyshev)
yfit = d + e * xnorm + f * ynorm

xfit = a + b * xnorm + c * ynorm  (legendre)
yfit = d + e * xnorm + f * ynorm

xnorm = (2 * x - (xmax + xmin)) / (xmax - xmin)
ynorm = (2 * y - (ymax + ymin)) / (ymax - ymin)

```

Xnorm and ynorm are the input x and y values normalized between -1.0 and 1.0.

2. For a higher order fit, say xorder=4 yorder=4 and xterms=1 (full), the format is more complicated. The second surface is computed by fitting the higher order surface to the residuals of the first fit. The geomap output will look something like the following.

```

surfacel      11
  surface(xfit) surface(yfit)      (surface type 1=cheb, 2=leg, 3=poly)
  xxorder(xfit) yxorder(yfit)      (always 2)
  xyorder(xfit) yyorder(yfit)      (always 2)
  xxterms(xfit) yxterms(yfit)      (always 0)
  xmin(xfit)    xmin(yfit)          (geomap input or data)
  xmax(xfit)    xmax(yfit)          (geomap input or data)
  ymin(xfit)    ymin(yfit)          (geomap input or data)
  ymax(xfit)    ymax(yfit)          (geomap input or data)
  a             d
  b             e
  c             f
surface2      24
  surface(xfit) surface(yfit)      (surface type 1=cheb, 2=leg, 3=poly)
  xxorder(xfit) yxorder(yfit)      (4)
  xyorder(xfit) yyorder(yfit)      (4)
  xxterms(xfit) yxterms(yfit)      (1 in this case)
  xmin(xfit)    xmin(yfit)          (geomap input or data)

```

xmax(xfit)	xmax(yfit)	(geomap input or data)
ymin(xfit)	ymin(yfit)	(geomap input or data)
ymax(xfit)	ymax(yfit)	(geomap input or data)
C00(xfit)	C00(yfit)	
C10(xfit)	C10(yfit)	
C20(xfit)	C20(yfit)	
C30(xfit)	C30(yfit)	
C01(xfit)	C01(yfit)	
C11(xfit)	C11(yfit)	
C21(xfit)	C21(yfit)	
C31(xfit)	C31(yfit)	
C02(xfit)	C02(yfit)	
C12(xfit)	C12(yfit)	
C22(xfit)	C22(yfit)	
C32(xfit)	C32(yfit)	
C03(xfit)	C03(yfit)	
C13(xfit)	C13(yfit)	
C23(xfit)	C23(yfit)	
C33(xfit)	C33(yfit)	

where the C_{mn} are the coefficients of the polynomials P_{mn} , and the P_{mn} are defined as follows

$P_{mn} = x ** m * y ** n$ (polynomial)

$P_{mn} = P_m(x_{norm}) * P_n(y_{norm})$ (chebyshev)

$P_0(x_{norm}) = 1.0$

$P_1(x_{norm}) = x_{norm}$

$P_{m+1}(x_{norm}) = 2.0 * x_{norm} * P_m(x_{norm}) - P_{m-1}(x_{norm})$

$x_{norm} = (2 * x - (x_{max} + x_{min})) / (x_{max} - x_{min})$

$P_0(y_{norm}) = 1.0$

$P_1(y_{norm}) = y_{norm}$

$P_{n+1}(y_{norm}) = 2.0 * y_{norm} * P_n(y_{norm}) - P_{n-1}(y_{norm})$

$y_{norm} = (2 * y - (y_{max} + y_{min})) / (y_{max} - y_{min})$

$P_{mn} = P_m(x_{norm}) * P_n(y_{norm})$ (legendgre)

$P_0(x_{norm}) = 1.0$

$P_1(x_{norm}) = x_{norm}$

$P_{m+1}(x_{norm}) = ((2m + 1) * x_{norm} * P_m(x_{norm}) - m * P_{m-1}(x_{norm})) / (m + 1)$

$x_{norm} = (2 * x - (x_{max} + x_{min})) / (x_{max} - x_{min})$

$P_0(y_{norm}) = 1.0$

$P_1(y_{norm}) = y_{norm}$

$P_{n+1}(y_{norm}) = ((2n + 1) * y_{norm} * P_n(y_{norm}) - n * P_{n-1}(y_{norm})) / (n + 1)$

$y_{norm} = (2 * y - (y_{max} + y_{min})) / (y_{max} - y_{min})$

Hopefully I have copied this all down correctly. The main points to remember is that the magnitudes of the coefficients reflect both the function type (polynomial, chebyshev, or legendre) and the normalization (x_{min} , x_{max} , y_{min} , y_{max}).