



CTF File Format

115.01.1609.FF

Sep 27, 2016

Columnar Table Format

The CODAR Columnar Table Format (CTF) is a text file for output data that is easy to read using MatLab or other applications and is easy to view in a text editor that doesn't automatically wrap long lines. It contains keyword descriptors and optional table or tables of columnar data.

Some key ideas for this format are:

1. **MatLab** compatible blind matrix reading.
2. **User readability.** Verbose Keywords and clean looking columnar data.
3. **Flexibility.** While some keywords are required by the CTF, most keywords are solely dependent on the file. Except where explicitly required, keyword order is typically unimportant.
4. **Expandability.** New keywords can be added/inserted without breaking existing readers. The columnar data can also be added to without breaking existing readers. If the key **%TableTypeColumnTypes:** is adhered to, then the columns can be rearranged or new columns can be inserted without breaking existing readers.

SeaSonde uses the CTF format for Radials, Ellipticals, Totals, Diagnostics, Tracks, Loops, and future Pattern files. Each of these files has its own file format document, which is built upon the CTF format.

- Text content. Roman ASCII encoding.
- Each text line may end with '\r' (return,char 13) or '\n' (newline,char 10) or '\r\n' or '\n\r'. If this is a problem for a particular platform, the file can be run through a converter utility, which exists already on most platforms. The maximum line length is 4096 characters, not counting the one or two end-of-line characters '\r\n'.
On OS X, the default the end-of-line is a '\n'.
- All non-table data lines start with '%' (percent) because MatLab blind matrix-reads ignore these lines. All tables data after the first table that are of a different type start with data with a '% ' (percent,space)

- **File Name Format Guide.**

This is not enforced and is meant for main product outputs so that they may be easily located, processed, archived, and moved between Mac, Unix, and other platforms.

- Case Insensitive. If used on a platform that has a case sensitive file system (like UNIX) the file can be renamed using a conversion utility.
- The purpose of the filename is to help software identify what to process and to help the user locate a file by name. It is not good programming practice to decode the filename for meta data like the site code or time stamp.
- No spaces in the file name are allowed.
- Minimum of 25 characters.

Formatted to: 'TTTS_XXXX_YYYY_MM_DD_HHMMSS.ctf'

TTTS is the type of file. Generally the TTT first three chars specify the file type and the last S specifies the style of processing or variation.

When looking for files, matching first three chars should be considered the same type.

'TOTL' for Total Vectors UV

'RDLi' for Radial Vectors UV from Merging algorithm with ideal antenna pattern

'ELTm' for Elliptical Vectors UV from Bistatic Processing with measured antenna pattern.

XXXX is the site code name. Four chars. No spaces. Must start with Alpha char. Followed by Alphanumeric.

YYYY is the year

MM is the two character month from 01 to 12

DD is the two character day from 01 to 31

HH is the two character 24 hour from 00 to 23

MM is the two character minute from 00 to 59

SS is the two character second from 00 to 59 (Only if seconds are needed.)

_ is the underscore separator character and is a must have. (don't use spaces or hyphens)

- '.ctf', is the default extension, but any extension may be used. A CTF file should be determined by reading the first ten lines looking for the %Filetype keyword and matching the type parameter to the kind of file you expect to find or are able to read.
- There may be extra characters after the minute field and before the '.ctf' extension. These characters should generally be ignored. These extra characters should not start with a number and typically should start with an under score '_'
- For radials and totals the time stamp represents the event center of the data.

Example: 'TOTL_MBay_2000_08_15_1200.tuv'

- **Keywords**

- Line starts with “%”, followed by keyword, followed by “:” followed by parameter data
- Keyword is NOT Case sensitive.
- Keyword does NOT contain spaces between the “%” and the “:” characters.
- Keyword is 1 to 32 alphanumeric characters (a-z,A-Z,0-9)
- Keyword should be verbose enough as to be able to guess at its meaning. (Don't use ambiguous abbreviations.)
- New Keywords should be registered with CODAR to prevent duplicate but different usage.
- When the keyword does not imply the data scale like (cm, km), the units should be kept in base units like Hz, meters, seconds, degrees or the units should follow the data like '24 MHz'
The actual meaning of the parameter data should be registered with the keyword.
- When processing, keywords, unless otherwise noted, line order typically does not matter and should not be counted upon if unless absolutely necessary.
- When processing, keywords are processed in order from first line in file to last line, which means that line order can matter. For example, when a reader reads the table it will need to know everything that pertains to that table before reading it.
- A keyword may appear as many times as necessary to show a change in the following data. Which also means that there may be multiple tables in a file. This means the CTF file is generally sequential in nature. Prevalent keyword data should come before the table data.

- **Parameter Data**

- Parameter Data follows each keyword after the ':'. A keyword need not have any parameters. A space is typically added after the ':' for visual clearness. The parameter(s) for keyword are dependent on the keyword and should not change units or order across files, file versions, and different file types.
- Parameters are separated by one or more spaces. A multi-word parameter (containing spaces) must be enclosed by “” quotes; otherwise just the next word separated by spaces is the parameter. The quotes should be removed by Reader software.
- Floating point numbers should have a decimal point '.' even if a whole value. They may be encoded in scientific format where E is used for 10^x 1000 is the same as 1E3 or 1.000E3. They might also be 'NAN' for not a number, '+INF' and '-INF' for plus, minus infinity.

- **Comments**

- All generic file comments start with ‘%%’ (two percent chars). Keywords can be disabled by inserting an extra ‘%’, turning ‘%’ into a ‘%%’ and the keyword into a comment.

- A '%' in the parameter part of a keyword line marks the beginning of a comment to the end of the line unless enclosed by "quotes"
- A line that start with '% ' (percent, space) that is not between a table start and table end keywords is a comment. Optional tables beyond the first table may have optionally have data rows that starts with a '% '(percent, space). The '% ' in these extra tables is to support MatLab blind matrix reads in which only the first table is implicitly understood.

- **Tables**

- Tables are optional. The file might be nothing but keywords. It's possible for the file to contain no tables, or that it may contain several tables of the same type or even different types.
- There must always be a keyword '%TableType <type> <subtype>' which describes the Table data type for each table before the %TableStart: keyword.
- While not strictly necessary, one should use a %TableColumnType: key which tags the data in each column. The use of the key allows table data to reordered and allows for future insertions and deletions of columns.
- For multiple tables, there must be a %TableStart: <table sequence#> and %TableEnd: keywords for each table.
- Table Data lines start with at least one ' ' (space) character or '% '(percent,space). The '% '(percent space is typically used on secondary tables that are of different type than the first table. This so that a MatLab blind matrix read will work on the main data and ignore any extra data in a different format. One example is the radial files where extra tables indicating the diagnostic information at the time of the radial are added.
- Data items/parameters are separated by one or more spaces. No tabs. No Commas. A multi-word item (contains spaces) must be enclosed by "" quotes
- Table data are file and usage type specific.

- **Blank Lines**

- No blank or empty lines. A CTF Writer should never add a blank line (use an empty comment '%' instead). A CTF Reader should ignore blank lines.

CTF Reading Notes

1. Don't assume file content by the name. Read the first ten lines and look for the %FileType: keyword. This will tell the contents of the CTF file. If this keyword is missing, then you should assume it is not a CTF file.
2. Ignore keywords you do not know or need.
3. Read each line sequentially and parse it independently without an expected keyword order. If you really need a keyword to precede another or table then you will need take extra precautions to ensure that you got your required keyword(s) before the current keyword or table data.
4. Allow for any type of line ending character(s) 0x10 linefeed, 0x13 carriage return, or combo.
5. Allow that there may be extra tables in the file. If you don't know about or desire the extra tables then you should ignore them (table keywords and rows).
6. Allow that when in a table (between %TableStart: and %TableEnd) the row data might start with '% ' (percent,space) or ' ' (space)
7. Take advantage of the %TableColumnTypes: keyword. This keyword contains a list of four character code descriptions of the data type in order for each table column. This allows the columns to be in any order and if you don't recognize a column type then you can skip it to process the next column. This allows for future modification and expansion without breaking your reader.
8. The MatLab blind matrix read is a quick way to get columnar data into an array. It does not read any line that starts with a '%', which means that you are assuming what the meta data is already and the table data layout. A better way is to write a MatLab CTF Reader specific to your needs.

CTF Writing Notes

1. The first line should be '%CTF: 1.0'.
2. The second line should typically (must be in the first ten lines) be the '%FileType: <type> <subtype>', where <type> is a unique four character identifier of the file. The <subtype> is a four character, which identifies variations in the file. A reader that recognizes the <type> should be able to understand most of the content regardless of the <subtype>.
3. The first table should contain row data that starts with a space. The following tables may optionally start row data with '% ' (percent space) to keep the main table compatible with Matlab read). The '% ' (percent space) is typically used on secondary tables that are of different type than the first table. This so that a MatLab blind matrix read will work on the main data and ignore any extra data in a different format. One example is the radial files where extra tables indicating the diagnostic information at the time of the radial are added.
4. Tables must be preceded by '%TableType: <type> <subtype>' so that the reader knows what the table is. It should also be preceded by %TableColumns: %TableRows: and %TableColumnTypes: so that the reader knows what to expect.
5. Before finishing, you should write a '%ProcessedTimeStamp: <yyyy> <mm> <dd> <hh> <mm> <ss>'. This is a better indicator of original processing time than the file modification date, which sometimes changes with file transfers or hand editing.
6. Before finishing, you should also write a list of the processing tools used upon the data with the keyword '%ProcessingTool: <name> <version>'.
7. The last line should be a '%End:'. While not strictly required, this marks that the file was successfully written and that the software did not break in the middle or some other unforeseen catastrophe.
8. When creating new keywords, be verbose. This file format is about clarity not compactness.

Known Keywords

%CTF: <version>

As of this document, the version is 1.00. It will change upon revision to this specification, which will require a CTF reader to parse the file differently. This will be the first keyword preceding even comments to indicate the nature of the file.

NOTE: Before Jan 31, 2005 this keyword did not exist, so you should allow for it to be missing at this time. Previously to this first version the CTF Spec called that '% ' (percent,space) always represented a comment, but none of the CODAR output files ever used this. This spec allows for '% ' to precede the row data for optional tables beyond the first table.

%FileType: <type> <subtype> <name>

This keyword should be the within the first ten keywords in the file. It describes the usage of the file.

<type> – file type. Four alphanumeric characters

Some example types in use by SeaSonde

LLUV Lon Lat U V Data

RBVD Radial Only Data

TRAC GPS Tracking Data

LOOP Antenna Pattern Measurement Data

<subtype>

Some example subtypes for LLUV

TOT3 Current (Total) Vectors map type3

RDL3 Radial Vectors map type 3

ELP1 Elliptical Vectors map type 1

<name> (Might be wrapped in quotes.)

Some Example naming

RadialMap Radial Vector File

CurrentMap Total Vector File

Elliptical Elliptical Radial Vector File

Transponder Transponder Tracking File

%TimeStamp: <yyyy> <mm> <dd> <hh> <mm> <ss>

Time stamp of data.

<yyyy> year 0001 to 9999

<mm> month 01 to 12

<dd> day 01 to 31

<hh> hour 00 to 23

<mm> minute 00 to 59

<ss> seconds 00 to 59

%TimeZone: <name> <offset> <daylight>

<name> three character name of timezone. (Might be wrapped in quotes.)

<offset> hours from UTC/GMT

<daylight> Daylight savings is in effect. 0(zero) if not. 1(one) if yes.

%TimeCoverage: <minutes>

<minutes> minutes that data spans.

%Site: <site> <name>

<site> Site Code. Four alphanumeric characters .

<name> Verbose name for site

%Origin: <lat> <lon>

Origin Location of on which vectors are base. It is important for converted from/to older rad/tot file formats. X,Y and Range,Bearing data are relative to this location.

This is the Receiver location for radials and the grid origin for totals

<lat> latitude in fractional degrees. Positive North. Negative South.

<lon> longitude in fractional degrees. Positive East. Negative West

%ReceiverLocation: <lat> <lon>

<lat> latitude in fractional degrees

<lon> longitude in fractional degrees

%TransmitterLocation: <lat> <lon>

<lat> latitude in fractional degrees

<lon> longitude in fractional degrees

%AntennaBearing: <deg>

<deg> Antenna loop1 direction in degrees clockwise from true north.

%TransmitCenterFreqMHz: <freq>

<freq> Transmitter frequency Center in MHz

%TransmitBandwidthKHz: <freq>

<freq> Transmitter frequency bandwidth in kHz (might be negative to indicate down sweep)

%RangeStart: <cells>

<cells> Spectra Range cell start of data (0 is at receiver location)

%RangeEnd: <cells>

<cells> Spectra Range cell end of data (0 is at receiver location)

Number of RangeCells is (%RangeEnd: - %RangeStart: + 1)

%RangeResolutionKMeters: <distance>
<distance> Distance between each range cell in kilometers

%DopplerCells: <cells>
<cells> Number of doppler cells used in processing.

%DopplerResolutionHzPerBin: <res>
<res> is the Hz per doppler cell used in processing.

%BraggPolarity: <pol>
<pol>
Both Both Postive & Negative Bragg are included
Positive Postive Bragg is included
Negative Negative Bragg is included

%BraggVelocity: <vel>
<vel> Theoretical position of the bragg peaks in cm/s from 0 doppler.

%NumberMergedRadials: <num>
<num> Number of source radials in radial merging process.

%MusicParameters: <a> <c>

%SmoothingParameters: <a> <c>

%SiteList: <site1> <site2> <site3> <site4> <site5> <site6> ...
SiteCodes that contributed to Totals.

%PhaseCorrections: <Phase1> <Phase2> ...

%AmplitudeCorrections: <Ampl1> <Ampl2>

%AntennaCorrection: <type> <source>
<type> Ideal/Measured
<source> filename used.

%MaximumCurrentVelocity: <max>
<max> Maximum allowed current filter in cm/s

%BraggHasSecondOrder: <yn>
<yn> Yes/No

%Owner: <name>

%Manufacturer: <name>

<name> string example "CODAR Ocean Sensors. SeaSonde."

%ProcessingTool: "<name>" <version>

<what> Software like "SpectaToRadial"

<ver> Version like "10.04"

%ProcessedTimeStamp: <yyyy> <mm> <dd> <hh> <mm> <ss>

%TableColumns: <num>

<num> number of parameters(columns) for each table line.

%TableColumnTypes: <fourcc> <fourcc>

This key contains an optional list of four character codes describing the data in each column. Without this key, the data's column order is assumed by the type of file. With this key, the data columns can be in any order or have some missing or have new columns. Readers need to match the column type code to the table data. If the reader doesn't understand the column code, then it should skip over that column.

<fourcc> a four character (alphanumeric) code to describe the data in each column.

Type Codes for LLUV files are:

LATD Latitude in fractional degrees. Positive East

LOND Longitude in fractional degrees. Positive North.

VELU U current velocity east component in cm/s.

VELV V current velocity north component in cm/s.

XDST Eastern Distance in km.

YDST Northern Distance in km.

RNGE Distance of vector from origin in km.

BEAR Bearing of vector in degrees clockwise from North.

VELO Current Velocity in cm/s.

HEAD Current Direction in degrees clockwise from North.

STDV Radial Quality/Sheer standard deviation in cm/s.

UQAL U component quality factor.

VQAL V component quality factor.

CQAL Covariance quality factor.

VFLG Grid Flag. Composite integer indicating points near coastline, baseline, ADCP marker and more.

S1CN Number of Radial vectors contributed by Site1 to total vector.

S2CN Number of Radial vectors contributed by Site2 to total vector.

S3CN Number of Radial vectors contributed by Site3 to total vector.

S4CN Number of Radial vectors contributed by Site4 to total vector.

S5CN Number of Radial vectors contributed by Site5 to total vector.

S6CN Number of Radial vectors contributed by Site6 to total vector.

%TableRows: <num>

The number of lines should be used as a helper for array allocations. It should be expected that the table might have been hand edited and therefore have deleted or additional table lines.

<num> number of table lines. 0 or more

%TableType: <type> <subtype>

<type> Table data type. Four alphanumeric characters .

LLUV lon, lat,u,v

Longitude in fractional degrees -180 to +180

Latitude in fractional degrees -90 to +90

U vector component in cm/s

V vector component in cm/s

RBVD range,bearing,velocity,direction

Range from **%Origin:** in km

Bearing from **%Origin:** to vector in degrees CW from True

Vector velocity in cm/s

Vector direction in degrees clockwise from True North

<subtype> Table data subtype. Four alphanumeric characters to describe extra columns

%TableStart: <table number> <table name>

Marks the beginning of the table data

<table number> optional table index number

<table name> optional table name

%TableEnd: <table number>

Marks the end of the table data.

<table number> is optional. If entered, must match %TableStart.

%Mark: <type>

Marks points of interest in the data

<type> type of marker followed by whatever data.

Example: '%Mark: RangeCell 3'

%GridSpacing: <distance kilometers>

%AveragingRadius: <distance kilometers>

%CoastLineAngles: <left> <right>

<left> lefthand bearing degrees NorthCW

<right> righthand (CW from left) bearing degrees NorthCW

%SiteIndex: <index>

Assigned Radial Site number

%RangeCellsToProcess: <cells>

Maximum Number of RangeCells to process into radials

%ReferenceAngle: <degreesCWN>

Baseline or Reference angle used to describe radial bearings.

%BraggSmoothingPoints: <points>

Number of doppler points used to smooth Bragg region for first order calculations.

%RadialBraggPeakDropOff: <dB>

dB down from peak to determine Bragg region for radial processing
'Header" file has this value as 'FactorDownPeakLimit'

%BraggHasSecondOrder: <yn>

Are second order waves expected to be present in Bragg region.
<yn> Yes/No

%WaveBraggPeakDropOff: <dB>

dB down from peak to determine Bragg region for wave processing

%WaveRangeCell: <cell>

Range Cell used for wave processing

%MaximumWavePeriod: <seconds>

Maximum wave period expected

%RadialBraggPeakNull: <dB>

dB used to search for second order separation in Bragg region for radial processing

%WaveBraggPeakNull: <dB>

dB used to search for second order separation in Bragg region for wave processing

%RadialBraggNoiseThreshold: <dB>

dB Bragg signal must be above the noise floor for Radial Processing

%WaveBraggNoiseThreshold: <dB>

dB Bragg signal must be above the noise floor for Wave Processing

%KeepLongTimeCrossSpectra: <numberOfFiles>
Number of CSA crossspectra files in the “Processed Files” folder to keep.

%KeepShortTimeCrossSpectra:
Number of CSS crossspectra files in the “Processed CSS Files” folder to keep.

%KeepRadialsAndWaves:
Number of Radial or Wave files in the “Processed CSS Files” folder to keep.

%PatternAmplitudeCorrections: <amp1> <amp2>
Amplitude correction factors used for antenna pattern correction

%BraggAmplitudeCorrections:
Amplitude correction factors measured from bragg information

%UsedAmplitudeCorrections:
Amplitude correction factors used for radial processing

%BraggPhaseCorrections:
Phase corrections used measure from bragg information

%UsePhaseCorrections:
Phase corrections used for radial processing

%CoastlineSector: <ang1> <ang2>
Visible coastline coverage sector in degrees CWN going clockwise from <ang1> to <ang2> inclusive.

%RadialMusicParameters: <1> <2> <3>
Radial MUSIC processing parameters

%RadialSmoothingParameters: <1> <2> <3> <4>
Radial smoothing parameters

Revision History

First Draft Sep 27, 2016

Copyright and Disclaimer

This document is copyrighted(c) by CODAR Ocean Sensors, Ltd and cannot be copied or reproduced in all or partial without expressed written consent by CODAR Ocean Sensors, Ltd.