SeaSonde Radial Site Release 6

CrossSpectra File Format V5

CrossSpectra files are produced by a SeaSonde Radial Site. They contain a snapshot in time of the ocean state in a cross spectra format, which is computed from nominally three antenna measurements. This data represents the reflected energy at each detectable range distance and radial doppler velocity as well as the cross spectra ratios of the antennas compared to each other. The cross spectra files are then used to calculate radial velocity vectors and ocean wave states.

The application SeaSondeAcquisition creates raw cross spectra in the "/Codar/SeaSonde/Data/Spectra/SpectraSeries/" folder. SeaSondeAcquisition saves the raw cross spectra file name as “CSQ_XXXX_YY_MM_DD_HHMMSS.cs” where XXXX is the site name; YY is the year, MM is the Month, HHMMSS is the 24hour/minute/second time.

The application CSPro reads the CSQ files and produces 'CSS_XXXX_YY_MM_DD_HHMM.cs' files in the "/Codar/SeaSonde/Data/Spectra/SpectraToProcess/" folder where XXXX is the site name; YY is the year, MM is the Month, HHMM is the 24hour and minute time. 'CSS' stands for CrossSpectra short time, which on a standard SeaSonde covers 15 minutes with an output time every 10minutes.

SeaSonde Spectra Processing also averages the CSS files and produces a 'CSA_XXXX_YY_MM_DD_HHMM.cs' normally every hour. 'CSA' stands for CrossSpectra Averaged. The 'CSA's are not normally used for radials and are used only for wave spectral processing.

The files are in binary format.
They have a variable size header section followed by the cross spectra products.
The data uses **Big-Endian** byte ordering (**M**ost **S**ignificant **B**yte first. This means that on Intel platforms, you will need to swap the byte order for the variable being read.)
IEEE floating point values single (4bytes) and double (8byte precision).
Two’s complement, integer values.

Data Type Definitions:

Uint8	Unsigned 8bit integer
Sint8	Signed 8bit integer
Uint16	Unsigned 16bit integer
Sint16	Signed 16bit integer
Uint32	Unsigned 32bit integer
Sint32	Signed 32bit integer
Uint64	Unsigned 64bit integer
Sint64	Signed 64bit integer
Float	IEEE single precision floating point number (4bytes)
Double	IEEE double precision floating point number (8bytes)
Size4	Unsigned 32bit integer indicating the size of following data
Char4	Four character code (meaning that the next four bytes make a four character string)
Char8	8byte string zero terminated (zero fill to get 8bytes max. must have at least one zero)
Char32	32byte string zero terminated (zero fill to get max. must have at least one zero)
Char64	64byte string zero terminated (zero fill to get max. must have at least one zero)
Char256	256byte string zero terminated (zero fill to get max. must have at least one zero)
Complex	2 IEEE single precision floating point numbers of real and imag pairs (8bytes, 4bytes each float)

Cross Spectra File Contents:

-Each File has two major sections. A **Header** section and a **CrossSpectra Data** section.

The **Header** section is as follows:

-The header is expandable. Each newer version also contains the information used the by older version.
-When reading a CrossSpectra file that is a newer version than you expect then use the Extent field to skip to the beginning of the cross spectra data.

-The following Header description is a set of data fields in order where each field description is a value type with implied size, followed by the field name, and followed by the field's description.

-All versions start with this as the very first bytes.

SInt16	nCsaFileVersion	File Version 1 to latest. *(If greater than 32, it's probably not a spectra file.)*
UInt32	nDateTime	TimeStamp. Seconds from Jan 1,1904 local computer time at site. The timestamp for CSQ files represents the **start time** of the data (nCsaKind = 1) The timestamp for CSS and CSA files is the **center time** of the data (nCsaKind = 2).
SInt32	nV1Extent	Header Bytes extension (Version 4 is +62 Bytes Till Data)

-Following is added info for version 2 to latest

SInt16	nCsKind	Type of CrossSpectra Data. 1 is self spectra for all used channels, followed by cross spectra. Timestamp is start time of data. 2 is self spectra for all used channels, followed by cross spectra, followed by quality data. Timestamp is **center time** of data.
SInt32	nV2Extent	Header Bytes extension (Version 4 is +56 Bytes Till Data)

- Following is added info for version 3 to latest

Char4	nSiteCodeName	Four character site code 'site'
SInt32	nV3Extent	Header Bytes extension (Version 4 is +48 Bytes Till Data)

-Note. If version is 3 or less, then nRangeCells=31, nDopplerCells=512, nFirstRangeCell=1

-Following is added info for version 4 to latest

SInt32	nCoverageMinutes	Coverage Time in minutes for the data. 'CSQ' is normally 5minutes (4.5 rounded) 'CSS' is normally 15minutes average. 'CSA' is normally 60minutes average.
SInt32	bDeletedSource	Was the 'CSQ' deleted by CSPro after reading.
SInt32	bOverrideSourceInfo	If not zero, CSPro used its own preferences to override the source 'CSQ' spectra sweep settings.
Float	fStartFreqMHz	Transmit Start Freq in MHz
Float	fRepFreqHz	Transmit Sweep Rate in Hz
Float	fBandwidthKHz	Transmit Sweep bandwidth in kHz
SInt32	bSweepUp	Transmit Sweep Freq direction is up if non zero, else down NOTE: CenterFreq is fStartFreqMHz + fBandwidthKHz/2 * -2^(bSweepUp==0)
SInt32	nDopplerCells	Number of Doppler Cells (nominally 512)
SInt32	nRangeCells	Number of RangeCells (nominally 32 for 'CSQ', 31 for 'CSS' & 'CSA')
SInt32	nFirstRangeCell	Index of First Range Cell in data from zero at the receiver. 'CSQ' files nominally use zero. 'CSS' or 'CSA' files nominally use one because CSPro cuts off the first range cell as meaningless.
Float	fRangeCellDistKm	Distance between range cells in kilometers.
SInt32	nV4Extent	Header Bytes extension (Version 4 is +0 Bytes Till Data)

		If zero then cross spectra data follows, but if this file were version 5 or greater then the nV4Extent would tell you how many more bytes the version 5 and greater uses until the data.

-Following is added info for version 5 to latest

SInt32	nOutputInterval	The Output Interval in Minutes.
Char4	nCreatorTypeCode	The creator application type code.
Char4	nCreatorVersion	The creator application version.
SInt32	nActiveChannels	Number of active antennas
SInt32	nSpectraChannels	Number antenna used in cross spectra
UInt32	nActiveChannelBits	Bit indicator of which antennas are in use msb is ant#1 to lsb #32
SInt32	nV5Extent	Header Bytes extension (Version 5 is +0 Bytes Till Data) If zero then cross spectra data follows, but if this file were version 6 or greater then the nV5Extent would tell you how many more bytes the version 6 and greater uses until the data.

-End of Header Section

-Begin Data Section:

The data section is a multi-dimensional array of self and cross spectra data.

Repeat For 1 to *nRangeCells*

Float[*nDopplerCells*] Antenna1 voltage squared amplitude self spectra.
Float[*nDopplerCells*] Antenna2 voltage squared amplitude self spectra.
Float[*nDopplerCells*] Antenna3 voltage squared amplitude self spectra.
*(**Warning**: Some Antenna3 amplitude values may be negative to indicate noise or interference at those doppler bins. These negative values should be **absoluted** before use.)*
Complex[*nDopplerCells*] Antenna 1 to Antenna 2 cross spectra.
Complex[*nDopplerCells*] Antenna 1 to Antenna 3 cross spectra.
Complex[*nDopplerCells*] Antenna 2 to Antenna 3 cross spectra.
if *nCsaKind* is 2 then also read or skip
Float[*nDopplerCells*] Quality array from zero to one in value.

End Repeat

-End Data Section

-End File

Note: To convert self spectra to dBm use:
10*log10(abs(voltagesquared)) - (-40. + 5.8)
-The -40. is conversion loss in the receiver and +5.8 is processing computational gain.