User Software :: CR-Tools :: The CalTable System

Usage of the CalTables

The CalTables are a system to store and retrieve calibration and other meta-data.

One piece of data (an atom) can be a string, a double, a dcomplex, an array of double or an array of dcomplex. Each atom is identified by its field name (which defines its type), the antenna ID, and the date for which it is valid.

A reading operation can look like this:

```
String tablename="lopes/calibration-data/CalTable/LOPES_CalTable";
CalTableReader table;
// open the table
table.AttachTable(tablename);
// define which value we are interested in
String fieldName="AntennaPosition";
Int AntennaID = 010101; //antenna 1 in LOPES notation
uInt date = time(); //get the value that is valid today
// get the data
Vector<Double> Pos;
if (!table.get(date, AntennaID, fieldName, &Pos)) {
  cerr << "Error while retrieving data" << endl;
};
cout << "The " << fieldName << " of antenna " << AntennaID <<
     " for the time label " << date << " is: " << Pos << endl;
```

Notes:

- In the `table.get()` call you have to supply the correct type for the field you are requesting. E.g. if the field whose `fieldName` you asked for is of the type `string`, then the variable in which the data is to be stored also has to be of the type `string`. Otherwise the call returns false.
- In the `table.get()` call, if the `fieldName` of `AntennaID` does not exist the call returns false.
- In the `table.get()` call, if no valid data for the requested time is found the call returns false.

Writing to the CalTable is only slightly more complicated, but still should only be done by trained personel. 

Adding fields to a CalTable should only be done by experts, as the later performance can be improved

by taking advantage of the internal structure of the CalTables.

Fields of the official LOPES CalTable

This is the list of fields in the official LOPES CalTable.

The definition of the fields is also in the description of the column in the master table.

Field Name	Data Type	Desctription
HWSetup	String	Status of the hardware setup. (Why has something in this subtable changed.)
Position	Vector<Double>	Relative antenna Position `[Northing, Easting, Height]` (in m) (Remark: This ordering is `[Y-axis, X-axis, Z-axis]`) Position `[0,0,0]` is the center of the KASCADE array.
FrequencyBand	Vector<Double>	Effective frequency band of this antenna `[StartFreq, StopFreq]` (in Hz)
Polarization	String	Polarization direction of this antenna/channel. (“`EW`” for East-West (standard) or “`NS`” nor North-South)
ElGainCal	Vector<Double>	Gain calibration values. Multiply to the frequency domain data to correct for the electronic gain. (I.e. the factor between FFT and CalFFT, so square this value before applying to power data.) Usually the values are positive, negative values correct for wrong orientation of the dipole.
ElGainCalFreq	Vector<Double>	Frequency axis for ElGainCal (in Hz). The vectors of ElGainCal and ElGainCalFreq have the same length, so the n-th value of ElGainCal belongs to the frequency stored in the n-th value of ElGainCalFreq
PhaseCal	Vector<DComplex>	Relative Phase calibration values. Multiply to the frequency domain data to correct for phase differences of the filters etc. Be careful during interpolation: ensure that `abs(PhaseCal)==1`
PhaseCalFreq	Vector<Double>	Frequency axis for PhaseCal (in Hz). Same structure a ElGainCalFreq for ElGainCal.
Delay	Double	Relative delay of the antenna (in seconds).
PhaseRefPhases	Vector<Double>	Phase calibration reference phases (in degrees). E.g. for the calibration on the TV transmitter.
PhaseRefFreqs	Matrix<Double>	Freqency ranges for the phase calibration `[ [StartFreq, StopFreq] , [<Band Number>] ]` (in Hz)
SampleJumps	Vector<Double>	Jumps by how many samples are to be tried in the phase calibration routine. (Only needed for the reference - currently the first (ID: 10101) - antenna.)
AntennaGainFaktor	Array<Double>	Array with the antenna gain faktors as they are eventually multiplied to the data: i.e. the value: `1/sqrt(Gain)` (with `Gain==Directivity`) with axes: [Frequency, Azimuth, Elevation]
AntennaGainFaktFreq	Vector<Double>	Frequency axis for AntennaGainFaktor (in Hz)

Field Name	Data Type	Desctription
AntennaGainFaktAz	Vector<Double>	Azimuth axis for AntennaGainFaktor 0=North, 90=East (in degrees)
AntennaGainFaktEl	Vector<Double>	Elevation axis for AntennaGainFaktor (in degrees)
AntennaAziGain	Array<Double>	The Antenna Gain Faktors for the azimuth polarization component. The value: `1/sqrt(Gain)` (with `Gain==Directivity`) with axes: [Frequency, Azimuth, Elevation] (Identical axes as AntennaGainFaktor.)
AntennaZeniGain	Array<Double>	The Antenna Gain Faktors for the zenith polarization component. The value: `1/sqrt(Gain)` (with `Gain==Directivity`) with axes: [Frequency, Azimuth, Elevation] (Identical axes as AntennaGainFaktor.)
AntennaAziPhase	Array<Double>	The Phase of the Antenna Gain for the azimuth polarization component. The phase angle in radians with axes: [Frequency, Azimuth, Elevation]
AntennaZeniPhase	Array<Double>	The Phase of the Antenna Gain for the zenith polarization component. The phase angle in radians with axes: [Frequency, Azimuth, Elevation]
PhaseRefAnt	Double	The ID of the reference antenna, to which phase differences for the phase calibration are measured. (Set to -1 to switch off phase calibration.)

Keywords of the official LOPES CalTable

This is the list of keywords in the LOPES CalTable, that can be read with `GetKeyword()`:

Keyword Name	Data Type	Desctription
minDate	Unsigned Int	Minumum value of the date field (=0)
maxDate	Unsigned Int	Maximum value of the date field (=0xffffffff = 2^{32}-1)
Observatory	String	Name of the Observatory for which this CalTable is Valid (=LOPES)
WWFilterS21	Vector<DComplex>	The S21 of the WainWright filter. (incl. phases)
WWFilterS21Freq	Vector<Double>	The frequency axis of the S21 of the WainWright filter.

(Internal) Structure of the CalTables

This is still missing. (And probably will for some time. Nagg me if you need it. Andreas H.)

← User Software • CR-Tools